

Prause, Felix; Hoppmann-Baum, Kai; Defourny, Boris; Koch, Thorsten

Article — Published Version

The maximum diversity assortment selection problem

Mathematical Methods of Operations Research

Provided in Cooperation with:

Springer Nature

Suggested Citation: Prause, Felix; Hoppmann-Baum, Kai; Defourny, Boris; Koch, Thorsten (2021) : The maximum diversity assortment selection problem, *Mathematical Methods of Operations Research*, ISSN 1432-5217, Springer, Berlin, Heidelberg, Vol. 93, Iss. 3, pp. 521-554,
<https://doi.org/10.1007/s00186-021-00740-2>

This Version is available at:

<https://hdl.handle.net/10419/286813>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<https://creativecommons.org/licenses/by/4.0/>



The maximum diversity assortment selection problem

Felix Prause¹ · Kai Hoppmann-Baum^{1,2} · Boris Defourny³ ·
Thorsten Koch^{1,2}

Received: 24 September 2020 / Revised: 15 April 2021 / Accepted: 16 April 2021 / Published online: 25 May 2021
© The Author(s) 2021

Abstract

In this article, we introduce the Maximum Diversity Assortment Selection Problem (MDASP), which is a generalization of the two-dimensional Knapsack Problem (2D-KP). Given a set of rectangles and a rectangular container, the goal of 2D-KP is to determine a subset of rectangles that can be placed in the container without overlapping, i.e., a feasible assortment, such that a maximum area is covered. MDASP is to determine a set of feasible assortments, each of them covering a certain minimum threshold of the container, such that the diversity among them is maximized. Thereby, diversity is defined as the minimum or average normalized Hamming distance of all assortment pairs. MDASP was the topic of the 11th AIMMS-MOPTA Competition in 2019. The methods described in this article and the resulting computational results won the contest. In the following, we give a definition of the problem, introduce a mathematical model and solution approaches, determine upper bounds on the diversity, and conclude with computational experiments conducted on test instances derived from the 2D-KP literature.

Keywords Combinatorial optimization · Mixed integer programming · Two-dimensional knapsack problem · Maximum diversity problem

✉ Felix Prause
prause@zib.de

Kai Hoppmann-Baum
hoppmann-baum@zib.de

Boris Defourny
defourny@lehigh.edu

Thorsten Koch
koch@zib.de

- ¹ AI in Society, Science and Technology, Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany
- ² Applied Algorithmic Intelligence Methods, Chair of Software and Algorithms for Discrete Optimization, Technische Universität Berlin, Str. des 17. Juni 135, 10623 Berlin, Germany
- ³ Department of Industrial and Systems Engineering, Lehigh University, 200 W Packer Ave, Bethlehem, PA 18015, USA

1 Introduction

The problem of packing rectangles into rectangular containers or to cut them from rectangular stock sheets arises in a variety of industrial applications. Thereby, one typically aims at determining a feasible solution where the wasted material or space is minimized. Consider for example the paper (Haessler 1971), glass (Hahn 1968), wood (Bouaine et al. 2018), or metal industry (Jakobs 1996). Here, rectangular pieces are needed for the production of certain goods, which are typically cut from given stock pieces (He et al. 2012). Another common application arising in logistics is to load pallets or containers (Huang and Chen 2007). Furthermore, similar problems appear in the production and operation of microchips, namely in the layout of processor chips (Wu and Chan 2005) and the dynamic allocation of memory as well as in multiprocessor scheduling (He et al. 2012). Finally, the editing and lay-outing of newspapers (Wei et al. 2009) or the arrangement of products on supermarket shelves (Problem description 2019) belong to this class of problems, too. They are typically modeled as some special variant of the two-dimensional Knapsack Problem (2D-KP), which we introduce in Sect. 2 and discuss in depth in Sect. 3.

However, for practitioners it is often useful when they are presented not only one optimal but a set of diverse “near-optimal” solutions from which she or he can choose. This holds especially for those tasks where it is hard to formalize or model important side constraints. Consider, for example, the last two examples from above. If a supermarket wants to investigate the buying behavior of its customers, an arrangement of the products minimizing the empty space is certainly desirable. Nevertheless, such a result on its own is not very meaningful for the particular task. Instead, the company needs to conduct tests with a variety of arrangements to assess whether they increase the purchasing rates or not (Kök et al. 2015). Similarly, when it comes to the layout of texts, pictures, or ads on a newspaper page, the result does not necessarily have to be minimal w.r.t. the resulting empty space, but has to come in some aesthetic appeal. Thus, presenting the user with a selection of assortments that cover some minimum threshold of the available area can be advantageous in all areas where experiences and subjective perceptions have an impact on the solution that is finally chosen.

Problems of this kind motivate the Maximum Diversity Assortment Selection Problem (MDASP), which we introduce in Sect. 2. Next, we review the literature on two inherent subproblems in Sect. 3: The above mentioned 2D-KP and the Maximum Diversity Problem (MDP), where a predefined number of elements has to be selected from a given set such that the diversity among them is maximized. Before discussing a mixed-integer quadratic programming (MIQP) model for MDASP in Sect. 5, we introduce MIQP formulations to determine upper bounds on the maximum diversity in Sect. 4. Furthermore, we present a generic two-stage heuristic in Sect. 6. Finally, we present extensive computational experiments in Sect. 7 and conclude with an outlook on future research in Sect. 8.

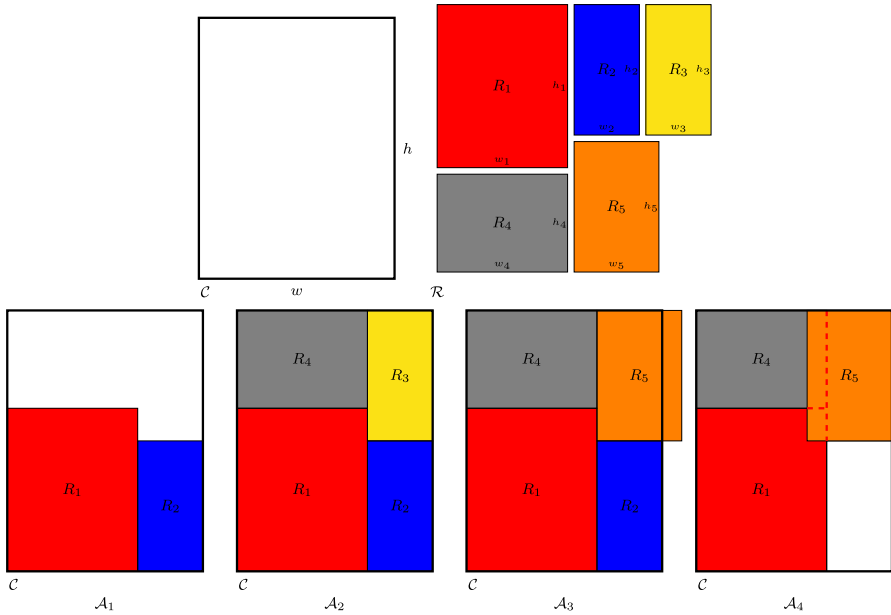


Fig. 1 Container \mathcal{C} with rectangle set \mathcal{R} . Assortment \mathcal{A}_1 is feasible and \mathcal{A}_2 is even optimal. On the other hand, assortments \mathcal{A}_3 and \mathcal{A}_4 are not feasible

2 Definitions and problem setup

For MDASP we are given a rectangle \mathcal{C} with width $w \in \mathbb{Z}_{\geq 0}$ and height $h \in \mathbb{Z}_{\geq 0}$, which we call *container*. Furthermore, we are given a set of rectangles $\mathcal{R} := \{R_1, \dots, R_n\}$ and each of them is associated with its width $w_i \in \mathbb{Z}_{\geq 0}$ and its height $h_i \in \mathbb{Z}_{\geq 0}$. Next, an *assortment* is a subset $\mathcal{A} \subseteq \mathcal{R}$ of rectangles, i.e., $\mathcal{A} \in \mathcal{P}(\mathcal{R})$ where $\mathcal{P}(\mathcal{R})$ denotes the powerset of \mathcal{R} . We call an assortment \mathcal{A} *feasible* if it can be placed in the container \mathcal{C} without overlapping, i.e., if we can assign a bottom-left corner coordinate $(x_i, y_i) \in \mathbb{R}^2$ to each rectangle $R_i \in \mathcal{A}$ such that $[x_i, x_i + w_i) \times [y_i, y_i + h_i) \subseteq [0, w) \times [0, h)$ and for all $R_i, R_j \in \mathcal{A}$ with $i \neq j$ we have $[x_i, x_i + w_i) \times [y_i, y_i + h_i) \cap [x_j, x_j + w_j) \times [y_j, y_j + h_j) = \emptyset$, see Iori et al. (2020). In the sequel, we denote the set of all feasible assortments by \mathcal{F} . Next, each assortment \mathcal{A} has an associated value $v(\mathcal{A}) = \sum_{R_i \in \mathcal{A}} w_i h_i$, which is the sum of the areas of the rectangles it contains. We call an assortment \mathcal{A}^* *optimal* if it is feasible and if $v(\mathcal{A}^*) \geq v(\mathcal{A})$ holds for every $\mathcal{A} \in \mathcal{F}$. Determining an optimal assortment is called the two-dimensional Knapsack Problem (2D-KP). Note that we otherwise allow an arbitrary placement of the rectangles inside the container, i.e., we do not impose any further conditions, and we do not allow the rotation of rectangles. An example can be found in Fig. 1.

For MDASP we are furthermore given a threshold value $\underline{v} \in [0, v(\mathcal{C})]$, with $v(\mathcal{C}) := wh$ denoting the area of the container, as well as a natural number $m \in \mathbb{N}$ with $m \geq 2$. We call an assortment \mathcal{A} *\underline{v} -good* if it is feasible and if $v(\mathcal{A}) \geq \underline{v}$, and we denote the set of \underline{v} -good assortments by $\mathcal{F}_{\underline{v}} \subseteq \mathcal{F}$. Furthermore, a *selection* is a multi-subset

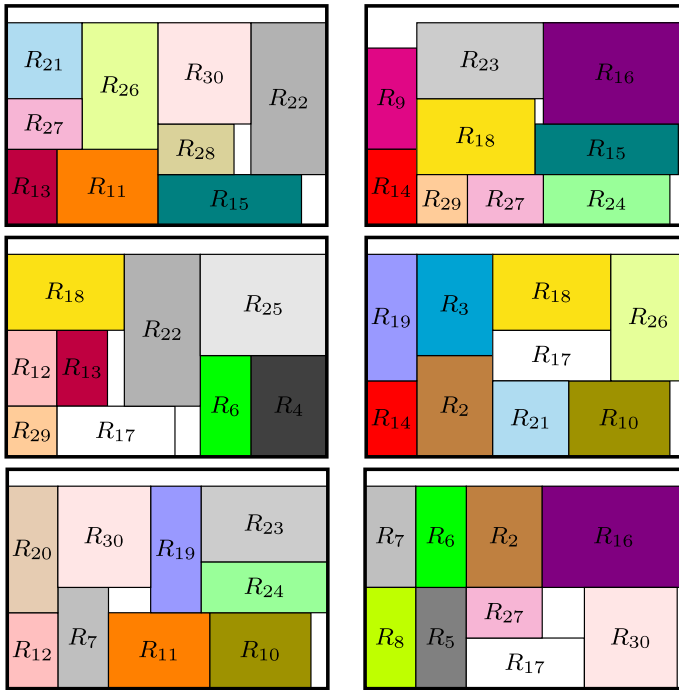


Fig. 2 Example for a selection of size $m = 6$ to the instance *leung8* w.r.t. δ_{min} , and a given threshold of $\underline{v} = 0.923$ which is the optimal value to the corresponding 2D-KP. The instance contains 30 rectangles, and the obtained selection has a diversity of 0.778, which was the most diverse selection that we obtained for this instance. Nevertheless, the best bound we found on this instance had a value of 0.916, compare with computational results in Sect. 7

\mathcal{S} of assortments, i.e., we allow that an assortment is contained more than once in \mathcal{S} , of cardinality $|\mathcal{S}| = m$. We call \mathcal{S} feasible if all of its assortments are \underline{v} -good, i.e., if $\mathcal{A} \in \mathcal{F}_{\underline{v}}$ for all $\mathcal{A} \in \mathcal{S}$. Additionally, we are given a diversity function δ for selections and call $\delta(\mathcal{S})$ the diversity of selection \mathcal{S} . The two diversity functions that we consider in this paper are based on the Hamming distances between assortment pairs contained in \mathcal{S} and are discussed in Sect. 2.1. Finally, a selection \mathcal{S}^* is called optimal if it is feasible and if $\delta(\mathcal{S}^*) \geq \delta(\mathcal{S})$ holds for all feasible selections \mathcal{S} . An example selection is shown in Fig. 2. In the following, we denote an MDASP instance \mathcal{I} as a quintuple $\mathcal{I} := (\mathcal{C}, \mathcal{R}, m, \underline{v}, \delta)$.

Lemma 1 *MDASP is NP-hard.*

Proof We prove this by reducing from 2D-KP, which is NP-hard, see Fekete and Schepers (2004) or Garey and Johnson (1979). Given an instance of 2D-KP with container \mathcal{C} and rectangle set \mathcal{R} , for arbitrary m and δ there exists a feasible assortment of value at least \underline{v} if and only if there exists a feasible selection for the MDASP instance $\mathcal{I} := (\mathcal{C}, \mathcal{R}, m, \underline{v}, \delta)$. □

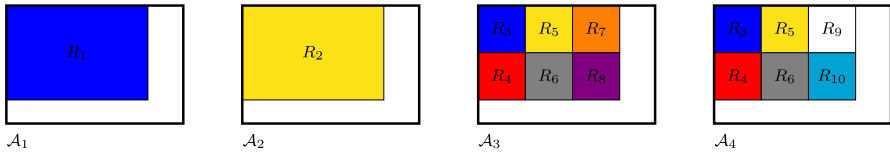


Fig. 3 The Hamming distance without normalization between \mathcal{A}_1 and \mathcal{A}_2 is 2, while the distance between \mathcal{A}_3 and \mathcal{A}_4 is 4. Using normalization, \mathcal{A}_1 and \mathcal{A}_2 have maximum distance 1, while the distance between \mathcal{A}_3 and \mathcal{A}_4 is $\frac{1}{3}$

2.1 Diversity of selections

A common distance measure applied to the subsets of a common superset is the *Hamming distance* (Hamming 1950). For two assortments \mathcal{A} and \mathcal{A}' , the (normalized) Hamming distance is defined as

$$d_H(\mathcal{A}, \mathcal{A}') := \frac{|\mathcal{A} \Delta \mathcal{A}'|}{|\mathcal{A}| + |\mathcal{A}'|} = \frac{|(\mathcal{A} \cup \mathcal{A}') \setminus (\mathcal{A} \cap \mathcal{A}')|}{|\mathcal{A}| + |\mathcal{A}'|},$$

where Δ represents the symmetric difference of sets. Furthermore, we define $d_H(\mathcal{A}, \mathcal{A}') = 0$ in case that $\mathcal{A} = \mathcal{A}' = \emptyset$. Note that we consider the normalized Hamming distance, i.e., we divide by $|\mathcal{A}| + |\mathcal{A}'|$, in order to ensure that the number of rectangles forming the assortments has no impact on the distance. An example demonstrating this rationale is depicted in Fig. 3.

Lemma 2 *Let $\mathcal{I} := (C, \mathcal{R}, m, \underline{v}, \delta)$ be an MDASP instance. For any two assortments $\mathcal{A} \subseteq \mathcal{R}$ and $\mathcal{A}' \subseteq \mathcal{R}$ we have $d_H(\mathcal{A}, \mathcal{A}') \in [0, 1]$.*

Based on the Hamming distance, we define two diversity functions for selections. Therefore, we denote the index set of the assortments of a selection by $M := \{1, \dots, m\}$ and introduce the *Minimum-Distance-Diversity* as

$$\delta_{min}(S) := \min_{\substack{i, j \in M \\ i < j}} \{d_H(\mathcal{A}_i, \mathcal{A}_j)\}$$

and the *Average-Distance-Diversity* as

$$\delta_{avg}(S) := \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^m d_H(\mathcal{A}_i, \mathcal{A}_j).$$

Lemma 3 *Let $\mathcal{I} := (C, \mathcal{R}, m, \underline{v}, \delta)$ be an MDASP instance and let S be a selection. Then $0 \leq \delta_{min}(S) \leq \delta_{avg}(S) \leq 1$.*

Proof

$$\begin{aligned}
\delta_{\min}(\mathcal{S}) &= \min_{\substack{i, j \in M \\ i < j}} \{d_H(\mathcal{A}_i, \mathcal{A}_j)\} \\
&= \frac{2}{m(m-1)} \sum_{k=1}^{m-1} \sum_{l=k+1}^m \min_{\substack{i, j \in M \\ i < j}} \{d_H(\mathcal{A}_i, \mathcal{A}_j)\} \\
&\leq \frac{2}{m(m-1)} \sum_{k=1}^{m-1} \sum_{l=k+1}^m d_H(\mathcal{A}_k, \mathcal{A}_l) = \delta_{\text{avg}}(\mathcal{S})
\end{aligned}$$

□

3 Related work and subproblems

MDASP has first been introduced in the 11th AIMMS-MOPTA Optimization Modeling Competition (AIMMS-MOPTA optimization 2019; Problem description 2019), which is part of the MOPTA conference series annually held at Lehigh University. To the best of our knowledge, there exists no previous work regarding it. Therefore, we give an overview of the literature concerning its two inherent subproblems instead: The two-dimensional Knapsack Problem (2D-KP) and the Maximum Diversity Problem (MDP).

As discussed in Sect. 1, the packing and cutting of rectangular items into or from rectangular containers arises in the context of many different applications. Thus, the 2D-KP is a topic that has been under investigation for a long time. According to Dowsland and Dowsland (1992), the first mathematical formulation of the problem was given by Kantorovich (1960) in 1939. Similar problem formulations were given by other authors during the 1950's as the work of Kantorovich was not translated until 1960.

For a general overview over 2D-KP, we refer to the survey papers (Cheng et al. 1994; Crainic et al. 2012; Dyckhoff 1990; Hinxman 1980; Hopper and Turton 2001), and in particular to the most recent one by Iori et al. (2020). Further, it is important to note that the two-dimensional Cutting Stock Problem (2CSP) is closely related to 2D-KP. In fact, solution algorithms for 2CSP can also be applied to 2D-KP. However, as there exists a variety of different variants of 2D-KP, we want to emphasize the work of Wäscher et al. (2007) and Lodi et al. (1999) regarding their classification. The former is based on the typology used in Dyckhoff (1990). Here, the different rectangular packing problems are identified as, e.g., Bin Packing, Knapsack, or Cutting Stock Problems, and additionally classified according to their dimensionality and objective as well as by the size, shape, and characteristics of the rectangles. On the other hand, the classification of Lodi et al. is based on the side constraints that need to be satisfied, e.g., if the rectangles can be placed freely in the container or are allowed to be rotated. Further, there exist a weighted and an unweighted variant of 2D-KP. In the weighted case each rectangle is assigned a certain value, and the objective of the problem is to maximize the sum of the values of all placed rectangles. In the unweighted case, the

value of each rectangle corresponds to its area. Thus, the objective here can either be seen as maximizing the area of all placed rectangles or as minimizing the empty or wasted space in the container. In this article we consider the unweighted variant of 2D-KP where the rectangles can be placed freely and are not allowed to be rotated.

In the literature, many mixed-integer programming (MIP) models exist for 2D-KP. For further details regarding MIP in general, we refer to Achterberg (2007). Hadjiconstantinou and Christofides (1995) introduced a formulation using the straightforward technique of discretizing the container. For each integer coordinate tuple in the container there exists a binary variable indicating if the corresponding point is already covered by a placed rectangle. Other MIP models featuring fewer variables and constraints using the relative positions of pairs of rectangles were introduced by Belov et al. (2009) and Egeblad and Pisinger (2009). The former approach is based on variables and corresponding constraints indicating whether two rectangles overlap when projected onto the x - or y -axis, from which at most one is allowed for a feasible assortment. The idea of the latter model is to use binary variables for each pair of rectangles to ensure that one of them is placed either over, under, left, or right of the other. Several other MIP formulations can be found in Beasley (1985), Gilmore and Gomory (1965), Hatefi (2017), Hifi (2001).

Furthermore, a variety of problem-specific *Branch-and-Bound* approaches has been developed (Christofides and Whitlock 1977; Clautiaux et al. 2007; Hifi and Zissimopoulos 1997). To improve their performance, Boschetti et al. (2002) present an upper bound, which can be used to significantly reduce the number of feasibility checks that have to be performed. Additionally, Fekete et al. (2007) introduce an approach to reduce the time that is needed for checking the feasibility of subsets of rectangles, i.e., whether it can be placed into the container and therefore forms a feasible assortment. They use graph structures to model equivalence classes of assortments and can determine if a certain rectangle subset is feasible or not by checking it for cycles and cliques. Their approach relies on a similar idea as the sequence-pair representation of rectangle-packings (Murata et al. 2003). Here, a possible packing is represented by two directed graphs encoding a sequence of the rectangles inside the container from left to right and from bottom to top, respectively. This idea was incorporated into solution approaches for 2D-KP, e.g., into the heuristics presented in Egeblad and Pisinger (2009), which extends (Pisinger 2007), where the representation is combined with a simulated annealing approach.

Next, we give a brief overview of the broad variety of heuristics and meta-heuristics that have been applied to 2D-KP. We start with deterministic algorithms, which are typically embedded in an iterative procedure applying different randomized orderings of the rectangles. The first type of heuristic used for 2D-KP are quasi-human algorithms that are inspired by the behavior of humans when solving a given problem. Consider, for example, the Least-Flexible-First algorithms of Wu et al. (2002), Wu and Chan (2005), and Huang and Chen (2007). The basic idea is to pack rectangles that are less flexible due to their size in the beginning, in order to have more flexibility when finishing up the packing. On the other hand, Wei et al. (2009) presented a Least-Waste-First heuristic where the rectangles are placed such that empty areas where no further rectangles can be placed are avoided. The third kind of placing procedures are so-called Best-Fit algorithms. In general, these heuristics are based on an evaluation

function. Here, the rectangles are not only chosen and placed in a way such that the resulting empty space is as small as possible, but they also have to fit “well” with respect to the already placed ones. Examples for this type of heuristic can be found in the work of He et al. (2012), de Armas et al. (2012), and in particular in the IBHP heuristic of Shiangjen et al. (2018). The last deterministic approach we mention here is the Dynamic Decomposition algorithm of Wang (2017). His idea is to sequentially decompose the container into smaller parts, pack them with rectangles, and rearrange them afterwards.

As mentioned above, several meta-heuristics have also been applied to 2D-KP. The approaches can roughly be partitioned into Greedy Randomized Adaptive Search Procedures (GRASPs) (MirHassani and Jalaeian Bashirzadeh 2015; Perboli et al. 2011; Álvarez Valdés et al. 2005), TABU Searches (Álvarez Valdés et al. 2007), Genetic Algorithms (Beasley 2004; Bortfeldt and Winter 2009), and Simulated Annealing approaches (Egeblad and Pisinger 2009; Leung et al. 2012). Further, there exist hybrid heuristics that combine deterministic algorithms and meta-heuristics (Gonçalves 2007; Gonçalves and Resende 2006; Hadjiconstantinou and Iori 2007).

The second problem, which is implicitly contained in MDASP, is to select a pre-defined number of elements from a given set such that the diversity among them is maximized. In our case this is the set of \underline{v} -good solutions $\mathcal{F}_{\underline{v}}$. This problem is known as the Maximum Diversity Problem or as Maximum Dispersion Problem (MDP) and is usually subdivided into the MAX-SUM and the MAX-MIN case. In the first case, the sum of the distances between the selected elements is maximized, while in the second case, one aims at maximizing the minimum distance between the chosen elements. This directly corresponds to our diversity measures δ_{avg} and δ_{min} .

Surveys on MDP have been published by Martí et al. (2013) and Sandoya et al. (2018). As for 2D-KP, there exists a variety of exact approaches to model and solve MDP, including MIP and IQP formulations (Ghosh 1996; Kuo et al. 1993) as well as special Branch-and-Bound approaches (Martí et al. 2010). Furthermore, different meta-heuristics have been used to tackle the problem, including for example GRASP heuristics (Resende et al. 2010; Silva et al. 2007, 2004), TABU Searches (Duarte and Martí 2007), and the Iterated Greedy Approach (Lozano et al. 2011). Finally, there also exist hybrid algorithms (Gallego et al. 2009; Santos et al. 2005) and greedy heuristics (Ravi et al. 1994).

4 Upper bounds on the diversity

In this section, we introduce two MIQP formulations to derive upper bounds on the maximum diversity of MDASP instances w.r.t. δ_{min} and δ_{avg} . The basic idea is to relax the problem by including assortments for which there may not exist a feasible placement in the container but that satisfy the \underline{v} -criterion, i.e., assortments contained in $\mathcal{G}_{\underline{v}} := \{\mathcal{A} \in \mathcal{P}(\mathcal{R}) \mid \underline{v} \leq v(\mathcal{A}) \leq v(\mathcal{C})\}$.

To simplify notation, we use an alternative expression for the Hamming distance in the following, namely

$$d_H(\mathcal{A}, \mathcal{A}') = \frac{|\mathcal{A} \Delta \mathcal{A}'|}{|\mathcal{A}| + |\mathcal{A}'|} = \frac{|\mathcal{A}| + |\mathcal{A}'| - 2|\mathcal{A} \cap \mathcal{A}'|}{|\mathcal{A}| + |\mathcal{A}'|} = 1 - \frac{2|\mathcal{A} \cap \mathcal{A}'|}{|\mathcal{A}| + |\mathcal{A}'|}.$$

4.1 Bounding the minimum-distance-diversity δ_{min}

Lemma 4 Let $\mathcal{I} := (\mathcal{C}, \mathcal{R}, m, \underline{v}, \delta_{min})$ be an instance of MDASP and let \mathcal{S} denote a feasible selection for \mathcal{I} . Then

$$\delta_{min}(\mathcal{S}) \leq 1 - \min_{\{\mathcal{A}_1, \dots, \mathcal{A}_m\} \subseteq \mathcal{G}_{\underline{v}}} \max_{\substack{i, j \in \{1, \dots, m\} \\ i < j}} \left\{ \frac{2|\mathcal{A}_i \cap \mathcal{A}_j|}{|\mathcal{A}_i| + |\mathcal{A}_j|} \right\}.$$

Proof From the definition of δ_{min} and since $\mathcal{F}_{\underline{v}} \subseteq \mathcal{G}_{\underline{v}}$ it follows that

$$\begin{aligned} \delta_{min}(\mathcal{S}) &\leq \max_{\{\mathcal{A}_1, \dots, \mathcal{A}_m\} \subseteq \mathcal{F}_{\underline{v}}} \min_{\substack{i, j \in \{1, \dots, m\} \\ i < j}} \{d_H(\mathcal{A}_i, \mathcal{A}_j)\} \\ &= \max_{\{\mathcal{A}_1, \dots, \mathcal{A}_m\} \subseteq \mathcal{F}_{\underline{v}}} \min_{\substack{i, j \in \{1, \dots, m\} \\ i < j}} \left\{ 1 - \frac{2|\mathcal{A}_i \cap \mathcal{A}_j|}{|\mathcal{A}_i| + |\mathcal{A}_j|} \right\} \\ &= \max_{\{\mathcal{A}_1, \dots, \mathcal{A}_m\} \subseteq \mathcal{F}_{\underline{v}}} \left\{ 1 - \max_{\substack{i, j \in \{1, \dots, m\} \\ i < j}} \left\{ \frac{2|\mathcal{A}_i \cap \mathcal{A}_j|}{|\mathcal{A}_i| + |\mathcal{A}_j|} \right\} \right\} \\ &= 1 - \min_{\{\mathcal{A}_1, \dots, \mathcal{A}_m\} \subseteq \mathcal{F}_{\underline{v}}} \max_{\substack{i, j \in \{1, \dots, m\} \\ i < j}} \left\{ \frac{2|\mathcal{A}_i \cap \mathcal{A}_j|}{|\mathcal{A}_i| + |\mathcal{A}_j|} \right\} \\ &\leq 1 - \min_{\{\mathcal{A}_1, \dots, \mathcal{A}_m\} \subseteq \mathcal{G}_{\underline{v}}} \max_{\substack{i, j \in \{1, \dots, m\} \\ i < j}} \left\{ \frac{2|\mathcal{A}_i \cap \mathcal{A}_j|}{|\mathcal{A}_i| + |\mathcal{A}_j|} \right\}. \end{aligned} \tag{1}$$

□

The following MIQP formulation UB_{min} determines expression (1), i.e., an optimal selection with respect to δ_{min} in $\mathcal{G}_{\underline{v}}$. Its variables and their meanings are listed in Table 1.

$$\max \quad 1 - z \tag{2}$$

$$\text{s.t.} \quad \sum_{R_i \in \mathcal{R}} w_i h_i c_{ia} \geq \underline{v} \quad \forall a \in M \tag{3}$$

$$\sum_{R_i \in \mathcal{R}} w_i h_i c_{ia} \leq v(\mathcal{C}) \quad \forall a \in M \tag{4}$$

$$\sum_{R_i \in \mathcal{R}} (c_{ia} + c_{ib}) = t_{ab} \quad \forall a, b \in M \text{ with } a < b \tag{5}$$

$$c_{ia} + c_{ib} \leq 1 + s_{iab} \quad \forall R_i \in \mathcal{R}, \quad \forall a, b \in M \text{ with } a < b \tag{6}$$

$$\sum_{R_i \in \mathcal{R}} s_{iab} \leq \frac{1}{2} z t_{ab} \quad \forall a, b \in M \text{ with } a < b \tag{7}$$

Table 1 Variables for MIQP model UB_{min}

| Variable | Description |
|-----------|--|
| c_{ia} | Binary variable indicating whether R_i is contained in \mathcal{A}_a |
| s_{iab} | Binary variable indicating whether R_i is contained in \mathcal{A}_a and \mathcal{A}_b |
| t_{ab} | Continuous variable equal to $ \mathcal{A}_a + \mathcal{A}_b $ |
| z | Continuous variable equal to the non-constant objective function value |

$$c_{ia} \in \{0, 1\} \quad \forall R_i \in \mathcal{R}, \quad \forall a \in M \tag{8}$$

$$s_{iab} \in \{0, 1\} \quad \forall R_i \in \mathcal{R}, \quad \forall a, b \in M \text{ with } a < b \tag{9}$$

$$t_{ab} \in [0, \dots, 2|\mathcal{R}|] \quad \forall a, b \in M \text{ with } a < b \tag{10}$$

$$z \in [0, 1] \tag{11}$$

Constraints (3) and (4) ensure that the generated assortments are contained in \mathcal{G}_v . Additionally, t_{ab} is equal to $|\mathcal{A}_a| + |\mathcal{A}_b|$ due to constraint (5). Further, we have $s_{iab} = 1$ if assortments \mathcal{A}_a and \mathcal{A}_b share rectangle R_i due to constraint (6). Therefore, z is equal to the maximum value of $\frac{2|\mathcal{A}_a \cap \mathcal{A}_b|}{|\mathcal{A}_a| + |\mathcal{A}_b|}$ among all pairs $a, b \in M$ with $a < b$ due to constraints (7) and because we minimize it in the objective function (2). Note that we intentionally choose the s_{iab} variables to be binary as the gap of the MIQP was tighter for a majority of the instances when the time limit was hit. This may be because the solver may not figure out that the variables are implicitly binary and thus benefits from the additional integrality conditions.

4.2 Bounding the average-distance-diversity δ_{avg}

The results from the previous subsection can also be adapted to δ_{avg} .

Lemma 5 *Let $(C, \mathcal{R}, m, \underline{v}, \delta_{avg})$ be an instance of MDASP and let \mathcal{S} denote a feasible selection. It holds that*

$$\delta_{avg}(\mathcal{S}) \leq 1 - \min_{\{\mathcal{A}_1, \dots, \mathcal{A}_m\} \subseteq \mathcal{F}_{\underline{v}}} \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^m \frac{2|\mathcal{A}_i \cap \mathcal{A}_j|}{|\mathcal{A}_i| + |\mathcal{A}_j|}.$$

Proof From the definition of δ_{avg} and since $\mathcal{F}_{\underline{v}} \subseteq \mathcal{G}_{\underline{v}}$ it follows that

$$\begin{aligned} \delta_{avg}(\mathcal{S}) &\leq \max_{\{\mathcal{A}_1, \dots, \mathcal{A}_m\} \subseteq \mathcal{F}_{\underline{v}}} \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^m d_H(\mathcal{A}_i, \mathcal{A}_j) \\ &= \max_{\{\mathcal{A}_1, \dots, \mathcal{A}_m\} \subseteq \mathcal{F}_{\underline{v}}} \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^m \left(1 - \frac{2|\mathcal{A}_i \cap \mathcal{A}_j|}{|\mathcal{A}_i| + |\mathcal{A}_j|} \right) \end{aligned}$$

$$\begin{aligned}
 &= \max_{\{\mathcal{A}_1, \dots, \mathcal{A}_m\} \subseteq \mathcal{F}_{\underline{v}}} \frac{2}{m(m-1)} \left(\frac{m(m-1)}{2} - \sum_{i=1}^{m-1} \sum_{j=i+1}^m \frac{2|\mathcal{A}_i \cap \mathcal{A}_j|}{|\mathcal{A}_i| + |\mathcal{A}_j|} \right) \\
 &= \max_{\{\mathcal{A}_1, \dots, \mathcal{A}_m\} \subseteq \mathcal{F}_{\underline{v}}} \left\{ 1 - \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^m \frac{2|\mathcal{A}_i \cap \mathcal{A}_j|}{|\mathcal{A}_i| + |\mathcal{A}_j|} \right\} \\
 &= 1 - \min_{\{\mathcal{A}_1, \dots, \mathcal{A}_m\} \subseteq \mathcal{F}_{\underline{v}}} \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^m \frac{2|\mathcal{A}_i \cap \mathcal{A}_j|}{|\mathcal{A}_i| + |\mathcal{A}_j|} \\
 &\leq 1 - \min_{\{\mathcal{A}_1, \dots, \mathcal{A}_m\} \subseteq \mathcal{G}_{\underline{v}}} \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^m \frac{2|\mathcal{A}_i \cap \mathcal{A}_j|}{|\mathcal{A}_i| + |\mathcal{A}_j|}. \tag{12}
 \end{aligned}$$

□

The following MIQP formulation UB_{avg} determines expression (12), i.e., an optimal selection w.r.t. δ_{avg} in $\mathcal{G}_{\underline{v}}$:

$$\max \quad 1 - \frac{2}{m(m-1)} \sum_{a=1}^{m-1} \sum_{b=a+1}^m z_{ab} \tag{13}$$

s.t. (3) – (6)

$$\sum_{R_i \in \mathcal{R}} s_{iab} = \frac{1}{2} z_{ab} t_{ab} \quad \text{for all } a, b \in M \text{ with } a < b \tag{14}$$

$$\begin{aligned}
 &\text{(8) – (10)} \\
 &z_{ab} \in [0, 1] \quad \text{for all } a, b \in M \text{ with } a < b. \tag{15}
 \end{aligned}$$

Most of the variables and constraints here are identical to the ones used in UB_{min} . However, here we introduce individual continuous variables (15) for each assortment pair \mathcal{A}_a and \mathcal{A}_b to determine their individual contributions using constraints (14) to the objective function (13).

4.3 Relations between diversity functions and their bounds

If we consider two MDASP instances that only differ by the diversity function, we can make the following observations.

Lemma 6 *Let $\mathcal{I}_1 = (\mathcal{C}, \mathcal{R}, m, \underline{v}, \delta_{min})$ and $\mathcal{I}_2 = (\mathcal{C}, \mathcal{R}, m, \underline{v}, \delta_{avg})$ be MDASP instances with optimal selections \mathcal{S}_1 and \mathcal{S}_2 , respectively. Then it holds that $\delta_{min}(\mathcal{S}_1) \leq \delta_{avg}(\mathcal{S}_2)$.*

Proof Applying Lemma 3 yields $\delta_{min}(\mathcal{S}_1) \leq \delta_{avg}(\mathcal{S}_1) \leq \delta_{avg}(\mathcal{S}_2)$. □

Corollary 1 *Let $\mathcal{I}_1 = (\mathcal{C}, \mathcal{R}, m, \underline{v}, \delta_{min})$ and $\mathcal{I}_2 = (\mathcal{C}, \mathcal{R}, m, \underline{v}, \delta_{avg})$ be two instances of MDASP. Then any upper bound on the diversity of \mathcal{I}_2 is an upper bound on the diversity of \mathcal{I}_1 as well.*

This result can directly be applied to our MIQP model UB_{avg} .

Corollary 2 Let $\mathcal{I} = (\mathcal{C}, \mathcal{R}, m, \underline{v}, \delta_{min})$ be an instance of MDASP and let \mathcal{S} denote a feasible selection. We have $\delta_{min}(\mathcal{S}) \leq UB_{avg}^*(\mathcal{I})$, where $UB_{avg}^*(\mathcal{I})$ is the optimal solution value for the corresponding MIQP model.

However, this bound cannot be tighter than the one we derive using UB_{min} .

Lemma 7 Let $\mathcal{I} = (\mathcal{C}, \mathcal{R}, m, \underline{v}, \delta_{min})$. We have $UB_{min}^*(\mathcal{I}) \leq UB_{avg}^*(\mathcal{I})$, where $UB_{min}^*(\mathcal{I})$ and $UB_{avg}^*(\mathcal{I})$ denote the optimal solution values for the corresponding MIQP models.

Proof Let $\mathcal{S}_1, \mathcal{S}_2$ be optimal selections for $UB_{min}(\mathcal{I})$ and $UB_{avg}(\mathcal{I})$, respectively. By Lemma 6 and by the optimality of \mathcal{S}_2 , it follows that

$$UB_{min}^*(\mathcal{I}) = \delta_{min}(\mathcal{S}_1) \leq \delta_{avg}(\mathcal{S}_2) = UB_{avg}^*(\mathcal{I}).$$

□

5 An MIQP model for MDASP

The rationale behind the following MIQP model for MDASP is the following. We construct a selection within $\mathcal{G}_{\underline{v}}$ by using either formulation UB_{min} or UB_{avg} , depending on the diversity function of the instance, see Sect. 4 for more details. However, for each assortment we additionally add the constraints of a MIP formulation for 2D-KP in order to ensure its feasibility, i.e., we guarantee that the selection is actually a subset of $\mathcal{F}_{\underline{v}}$ and therefore feasible itself. In the following example formulation (P) for δ_{min} , we use the inequalities of the MIP model of Egeblad and Pisinger (2009). It features the variables listed in Table 2.

$$(P) \quad \max \quad 1 - z \tag{16}$$

$$\text{s.t.} \quad \sum_{R_i \in \mathcal{R}} w_i h_i c_{ia} \geq \underline{v} \quad \forall a \in M \tag{17}$$

$$\sum_{R_i \in \mathcal{R}} w_i h_i c_{ia} \leq wh \quad \forall a \in M \tag{18}$$

$$\sum_{R_i \in \mathcal{R}} (c_{ia} + c_{ib}) = t_{ab} \quad \forall a, b \in M \text{ with } a < b \tag{19}$$

$$c_{ia} + c_{ib} \leq 1 + s_{iab} \quad \forall R_i \in \mathcal{R}, \forall a, b \in M \text{ with } a < b \tag{20}$$

$$\sum_{R_i \in \mathcal{R}} s_{iab} \leq \frac{1}{2} z t_{ab} \quad \forall a, b \in M \text{ with } a < b \tag{21}$$

$$l_{ija} + r_{ija} + u_{ija} + o_{ija} \geq c_{ia} + c_{ja} - 1 \quad \forall R_i, R_j \in \mathcal{R} \text{ with } i < j, \forall a \in M \tag{22}$$

$$x_{ia} - x_{ja} + w l_{ija} \leq w - w_i \quad \forall R_i, R_j \in \mathcal{R} \text{ with } i < j, \forall a \in M \tag{23}$$

$$x_{ja} - x_{ia} + w r_{ija} \leq w - w_j \quad \forall R_i, R_j \in \mathcal{R} \text{ with } i < j, \forall a \in M \tag{24}$$

$$y_{ia} - y_{ja} + h u_{ija} \leq h - h_i \quad \forall R_i, R_j \in \mathcal{R} \text{ with } i < j, \forall a \in M \tag{25}$$

Table 2 Variables for MIQP model (P) for MDASP

| Variable | Description |
|-----------|--|
| c_{ia} | Binary variable indicating whether R_i is contained in \mathcal{A}_a |
| s_{iab} | Binary variable indicating whether R_i is contained in \mathcal{A}_a and \mathcal{A}_b |
| t_{ab} | Continuous variable equal to $ \mathcal{A}_a + \mathcal{A}_b $ |
| z | Continuous variable equal to the non-constant objective function value |
| x_{ia} | Continuous variable determining the x-position of R_i in \mathcal{A}_a |
| y_{ia} | Continuous variable determining the y-position of R_i in \mathcal{A}_a |
| l_{ija} | Binary variable whether R_i is located left of R_j in \mathcal{A}_a |
| r_{ija} | Binary variable whether R_i is located right of R_j in \mathcal{A}_a |
| u_{ija} | Binary variable whether R_i is located under R_j in \mathcal{A}_a |
| o_{ija} | Binary variable whether R_i is located over R_j in \mathcal{A}_a |

$$y_{ja} - y_{ia} + h o_{ija} \leq h - h_j \quad \forall R_i, R_j \in \mathcal{R} \text{ with } i < j, \quad \forall a \in M \tag{26}$$

$$0 \leq x_{ia} \leq w - w_i \quad \forall R_i \in \mathcal{R}, \quad \forall a \in M \tag{27}$$

$$0 \leq y_{ia} \leq h - h_i \quad \forall R_i \in \mathcal{R}, \quad \forall a \in M \tag{28}$$

$$l_{ija}, r_{ija}, u_{ija}, o_{ija} \in \{0, 1\} \quad \forall R_i, R_j \in \mathcal{R} \text{ with } i < j, \quad \forall a \in M \tag{29}$$

$$c_{ia} \in \{0, 1\} \quad \forall R_i \in \mathcal{R}, \quad \forall a \in M \tag{30}$$

$$s_{iab} \in \{0, 1\} \quad \forall R_i \in \mathcal{R}, \quad \forall a, b \in M \text{ with } a < b \tag{31}$$

$$t_{ab} \in [0, \dots, 2|\mathcal{R}|] \quad \forall a, b \in M \text{ with } a < b \tag{32}$$

$$z \in [0, 1] \tag{33}$$

The objective function (16), constraints (17)–(21) and variables (30)–(33) correspond to model UB_{min} and construct a selection with maximum Minimum-Distance-Diversity in \mathcal{G}_v . On the other hand, constraints (22)–(28) and variables (27)–(29) originate from the MIP formulation of Egeblad and Pisinger (2009) for 2D-KP and ensure that the contained assortments are actually feasible. Thereby, constraint (22) ensures that if rectangles R_i and R_j are used in assortment \mathcal{A}_a , i.e., $c_{ia} = c_{ja} = 1$, then at least one of the four variables l_{ija} , r_{ija} , u_{ija} , or o_{ija} has to be equal to 1. This implies that R_i has to be placed left of R_j (23), right of R_j (24), under R_j (25), or over R_j (26), which guarantees that the two rectangles do not overlap. Furthermore, by the definition of the positioning variables x_{ia} and y_{ia} , see (27) and (28), each rectangle is placed within the container. Note that the constraints ensuring the feasibility of the assortments, i.e., (22)–(28), are similar to the constraints in the formulation of Padberg (2000).

5.1 A benders decomposition algorithm

Next, we describe a Benders decomposition approach for the introduced MIQP formulation for MDASP. For details regarding Benders decompositions, we refer to Benders (2005) and Geoffrion (1972). To derive it, we subdivide the model into its two sub-problems. The higher-level problem consists in constructing a diverse selection of

Algorithm 1 Generic Two-Stage Algorithm

Input: MDASP instance $\mathcal{I} = (\mathcal{C}, \mathcal{R}, m, \underline{v}, \delta)$
Output: Feasible Selection \mathcal{S} or UNSUCCESSFUL

- 1: $\mathcal{F}_s \leftarrow$ Sample feasible assortments \mathcal{A} for 2D-KP instance \mathcal{C}, \mathcal{R} with $v(\mathcal{A}) \geq \underline{v}$
- 2: **if** $\mathcal{F}_s = \emptyset$ **then**
- 3: **return** UNSUCCESSFUL
- 4: **else**
- 5: **return** Solution of MDP algorithm for instance \mathcal{F}_s, m, δ

assortments in $\mathcal{G}_{\underline{v}}$. The lower-level problems ensure the feasibility of the contained assortments. Thus, in our case the higher-level problem is UB_{min} or UB_{avg} , depending on the diversity function, and the lower-level problem is any MIP formulation or exact approach for 2D-KP to check the feasibility of the single assortments, e.g., the variables and constraints from Egeblad and Pisinger (2009) in example (P). If a solution for the higher-level problem has been found, but an assortment \mathcal{A} is identified as infeasible by a lower-level problem, corresponding no-good-cuts

$$\sum_{R_i \in \mathcal{A}} x_{ia} \leq |\mathcal{A}| - 1, \forall a \in M$$

are added to the higher-level problem, which is then solved again. These cuts ensure that no assortment containing \mathcal{A} as a subset is considered as feasible by the higher-level problem. Note that this separation problem, i.e., the lower-level problem, is NP-hard itself as we need solve an instance of 2D-KP.

6 A generic two-stage heuristic

Next, we present a generic two-stage heuristic for MDASP, see Algorithm 1. In its first stage, we use any heuristic or exact solution approach for 2D-KP to sample the space of \underline{v} -good assortments. We denote this sample set by $\mathcal{F}_s \subseteq \mathcal{F}$ in the following. Afterwards, we consider this subset in any exact or heuristic solution approach for MDP in order to determine a feasible selection of size m with respect to the diversity measure δ . Note that unless we are able to sample the complete set of \underline{v} -good assortments, i.e., $\mathcal{F}_{\underline{v}}$, and do apply an exact MDP approach, the algorithm does not necessarily determine an optimal solution.

For many MDP approaches from the literature the distances between the assortments have to be known prior to their execution. However, depending on the size of \mathcal{F}_s , determining them can be quite time-consuming. Thus, we introduce a new heuristic for MDP that does not rely on the prior availability of these distances.

Our algorithm, which is stated as Algorithm 2, is based on the idea of a random exchange, i.e., we start with a selection \mathcal{S}_b of m randomly chosen assortments and then iteratively check if a complete or partial exchange with another k assortments increases the diversity of the selection. A similar idea was suggested by Ghosh (1996). However, in his approach the exchange is based on an evaluation of all assortments. We avoid this by selecting the assortments completely at random and only determine the distances

Algorithm 2 MDP Random Exchange Heuristic

Input: $\mathcal{F}_S, m, \delta, \Delta$
Output: Selection \mathcal{S}_b

- 1: $\mathcal{S}_b \leftarrow$ Randomly choose m assortments $\mathcal{A}_1, \dots, \mathcal{A}_m \in \mathcal{F}_S$
- 2: $\delta_b \leftarrow \delta(\mathcal{S}_b)$
- 3: $k \leftarrow 1$
- 4: **While** $k \leq \Delta$
- 5: Choose $\mathcal{S}_r \subseteq \mathcal{F}_S \setminus \mathcal{S}_b$ with $|\mathcal{S}_r| = \lceil \frac{k}{100} \rceil$ at random
- 6: $\mathcal{S}_c \leftarrow \mathcal{S}_b \cup \mathcal{S}_r$
- 7: Determine $d_H(\mathcal{A}_i, \mathcal{A}_j)$ for all $\mathcal{A}_i, \mathcal{A}_j \in \mathcal{S}_c$ with $i < j$
- 8: $\mathcal{S}_b \leftarrow$ Solution of exact MDP approach for \mathcal{S}_c, m, δ
- 9: **If** $\delta(\mathcal{S}_b) > \delta_b$ **then**
- 10: $\delta_b \leftarrow \delta(\mathcal{S}_b)$
- 11: $k \leftarrow 1$
- 12: **Else**
- 13: $k \leftarrow k + 1$
- 14: **EndIf**
- 15: **EndWhile**
- 16:
- 17: **Return** \mathcal{S}_b

between the considered $m + k$ assortments in \mathcal{S}_c . Afterwards, we use an exact MDP formulation, depending on the diversity function that should be maximized, to choose m assortments from \mathcal{S}_c with maximum diversity, see Kuo et al. (1993) for example. Note that the diversity cannot decrease. The number k of assortments considered for an exchange increases with every 100 iterations that did not lead to an increase of the diversity, see line 5 of the algorithm. This count is reset whenever a more diverse selection is found. The idea here is, in particular when considering δ_{min} , that the diversity of the selection may depend on distances between multiple assortments. In this case, the exchange of only one assortment does not lead to an increase in the diversity. Thus, it is necessary to consider the replacement of more than one assortment at once. Algorithm 2 terminates after Δ unsuccessful iterations.

7 Computational experiments

In this section, we report on the results of our computational experiments. We conducted them using the two instances from the MOPTA competition (Problem description 2019) as well as modified 2D-KP and 2CSP instances, which are widely known from the literature. We evaluate the results for directly solving an instantiation of the MIQP formulation and when applying the Benders approach, which were both presented in Sect. 5, and for an instantiation of our generic two-stage heuristic from Sect. 6. We compare the three approaches w.r.t. the diversity of the best solution they determined. Additionally, we present the results of our upper bound computations using the formulations described in Sect. 4.

Before doing this, we investigate different exact and heuristic approaches for 2D-KP regarding the best generated solution and the total number of generated assortments. This is necessary in order to decide which MIP formulation to use within the MIQP

model and which heuristic to employ in the first stage of the heuristic. For the latter we are particularly interested in the number of generated assortments that satisfy the \underline{v} -criterion.

For our experiments, we considered $\underline{v} = (1 - \varepsilon)v^*$ with $\varepsilon = 0.05$ as threshold. Here, v^* denotes the best solution value which we determined during the 2D-KP runs. Thus, obtaining good solutions for 2D-KP obviously is a crucial task as the value of the best solution found by any of the approaches serves as the threshold value for all successive computational experiments.

7.1 Computational setup

All heuristic algorithms for 2D-KP were implemented in Ada 2012 using the GNAT Pro 19 compiler (AdaCore: GNAT 2019) and were run on an Intel(R) Xeon(TM) E5-2690 v4 CPU with 2.60GHz, four cores, and 32 GB RAM. The Benders approach was coded with Python v3.6 (Python Software Foundation 2020), and for the MIP and MIQP models Gurobi v9.0 (Gurobi Optimization 2019) was used as solver. For all computations, we set a time limit of 3,600 seconds. Additionally, for the computation of the upper bounds the focus of Gurobi was set to primarily improve the bounds.

7.2 Test instances

Since MDASP is a novel optimization problem, an important task was to come up with test instances. Before explaining how we derived test instances using 2D-KP and 2CSP instances, we first of all explain how the two test instances for the MOPTA competition were created.

7.2.1 Generation of MOPTA instances

For the AIMMS-MOPTA competition, data generation procedures were devised to produce problems of any size, that could exhibit some variety in the shape of the rectangles, as measured by the aspect ratio (height-width ratio), and in the size of the rectangles, as measured by the surface.

The generation procedure for the first data set has seven parameters $(n, w_{\min}, w_{\max}, \theta_1, h_{\min}, h_{\max}, \theta_2)$. It first generates samples w_i and h_i of real-valued random variables \mathcal{W}_i and \mathcal{H}_i representing the width and height of rectangle i for $i = 1, \dots, n$. \mathcal{W}_i and \mathcal{H}_i follow independent bounded power law distributions with shape parameters θ_1 and θ_2 respectively, restricted to the domain $[w_{\min}, w_{\max}] \times [h_{\min}, h_{\max}]$:

$$\mathbb{P}(w \leq \mathcal{W}_i \leq w + dw, h \leq \mathcal{H}_i \leq h + dh) \propto w^{\theta_1} h^{\theta_2} dw dh + o(dw dh)$$

$$\text{for } w \in [w_{\min}, w_{\max}), h \in [h_{\min}, h_{\max}). \quad (34)$$

This can be done by drawing some u_i, v_i uniformly in $[0, 1]$ and setting

$$w_i = \left((w_{\max}^{\theta_1+1} - w_{\min}^{\theta_1+1})u_i + w_{\min}^{\theta_1+1} \right)^{1/(\theta_1+1)}, \tag{35}$$

$$h_i = \left((h_{\max}^{\theta_2+1} - h_{\min}^{\theta_2+1})v_i + h_{\min}^{\theta_2+1} \right)^{1/(\theta_2+1)}. \tag{36}$$

The real-valued samples are then rounded up to the next integer. The data set was generated using $n = 200, w_{\min} = 40, w_{\max} = 200, \theta_1 = 1.8, h_{\min} = 40, h_{\max} = 200, \theta_2 = 0.8$. The container had width 300 and height 400. Having $\theta_1, \theta_2 > 0$ means that large rectangles are favored over small rectangles in the generation process.

The generation procedure for the second data set has seven parameters $(n, s_{\min}, s_{\max}, \theta_3, \ell_{\min}, \ell_{\max}, \theta_4)$. It first generates samples s_i and ℓ_i of real-valued random ratios \mathcal{S}_i and lengths $\mathcal{L}_i, i = 1, \dots, n$, following independent bounded power law distributions with shape parameters θ_3 and θ_4 respectively, restricted to the domain $[s_{\min}, s_{\max}] \times [\ell_{\min}, \ell_{\max}]$, defined similarly to (34):

$$\mathbb{P}(s \leq \mathcal{S}_i \leq s + ds, \ell \leq \mathcal{L}_i \leq \ell + d\ell) \propto s^{\theta_3} \ell^{\theta_4} ds d\ell + o(ds d\ell)$$

for $s \in [s_{\min}, s_{\max}), \ell \in [\ell_{\min}, \ell_{\max})$.

It then derives width and height samples w_i, h_i using a conditional rule: if $s_i \geq 1$, set $w_i = \ell_i/s_i$ and $h_i = \ell_i$ (tall rectangles); if $s_i < 1$, set $w_i = \ell_i$ and $h_i = s_i \ell_i$ (flat rectangles). The values are then rounded up to the next integer. The data set was generated using $n = 40, s_{\min} = 0.25, s_{\max} = 2, \theta_3 = 0.5, \ell_{\min} = 20, \ell_{\max} = 200, \theta_4 = 1$. The container was a square of sides of length 500.

The two data generation procedures are not equivalent. It can be checked that when $\mathcal{W}_i, \mathcal{H}_i$ follow truncated power laws, the distribution of the products $\mathcal{W}_i \mathcal{H}_i$ or the ratios $\mathcal{H}_i/\mathcal{W}_i$ do not themselves follow power laws. Thus, the two generation procedures control the distribution of the aspect ratios and surfaces in two different ways.

For both data sets, the parameters were determined after some tuning to make sure the problems were sufficiently challenging. This was done by estimating the computational time needed to obtain a pool of ϵ -optimal solutions to the two-dimensional knapsack problem formulated following (Hadjiconstantinou and Christofides 1995). To estimate the times, simplified instances were solved, obtained by scaling down by a factor 50 the dimensions of the container and rectangles and rounding them up to the next integer. Scale-and-round was used to reduce the number of binary variables needed to formulate the problems while hopefully preserving the relative degree of difficulty among the generated problems. The generation of a solution pool takes more time than the generation of a single solution but was deemed useful to measure the complexity of describing the set of good solutions from which a maximally-diverse solution is subsequently selected.

7.2.2 Derivation of instances from 2D-KP and 2CSP

As mentioned, we additionally created new MDASP instances based on 2D-KP and 2CSP instances, of which plentiful exist in the literature. In particular, we used the

Table 3 Packages of test instances given in the literature and by the AIMMS-MOPTA Competition. $|\mathcal{R}|_{min}$ is the smallest number of rectangles among all instances of the considered package, while $|\mathcal{R}|_{max}$ is the maximum cardinality of \mathcal{R} among all instances of the package

| Package | # Instances | $ \mathcal{R} _{min}$ | $ \mathcal{R} _{max}$ | Source |
|---------------|-------------|-----------------------|-----------------------|---|
| <i>2dcsp</i> | 60 | 20 | 50 | Imahori (2019) |
| <i>AH</i> | 360 | 1000 | 1000 | Bortfeldt and Gehring (2006) |
| <i>area</i> | 6 | 33 | 500 | Imahori (2019) |
| <i>beng</i> | 10 | 20 | 200 | Bengtsson (1982) |
| <i>Burke</i> | 13 | 10 | 3152 | Burke et al. (2004) |
| <i>bwmv</i> | 500 | 20 | 100 | Berkey and Wang (1987), Martello and Vigo (1998) |
| <i>C</i> | 21 | 16 | 197 | Hopper and Turton (2001) |
| <i>cgcut</i> | 3 | 16 | 62 | Christofides and Whitlock (1977) |
| <i>CX</i> | 7 | 50 | 15,000 | Pinto and Oliveira (2005) |
| <i>gcut</i> | 13 | 10 | 50 | Beasley (1985) |
| <i>HC</i> | 6 | 7 | 21 | Hadjiconstantinou and Christofides (1995) |
| <i>leung</i> | 10 | 10 | 50 | Leung et al. (2003) |
| <i>N</i> | 35 | 17 | 197 | Hopper (2000) |
| <i>ngcut</i> | 12 | 7 | 22 | Beasley (1985) |
| <i>Nice</i> | 6 | 25 | 1000 | Wang and Valenzela (2001) |
| <i>Nice36</i> | 33 | 100 | 5000 | Wang and Valenzela (2001) |
| <i>OKP</i> | 5 | 30 | 97 | Fekete and Schepers (2000) |
| <i>others</i> | 4 | 40 | 200 | AIMMS-MOPTA Competition Problem description (2019), Babu and Babu (1999) and Wang Wang (1983) |
| <i>Path</i> | 6 | 25 | 1000 | Wang and Valenzela (2001) |
| <i>Path36</i> | 33 | 25 | 5000 | Wang and Valenzela (2001) |
| <i>PB</i> | 5 | 10 | 20 | Lai and Chan (1997) |
| <i>T</i> | 35 | 17 | 199 | Hopper (2000) |
| <i>ZDF</i> | 16 | 580 | 75,032 | Shiangjen et al. (2018) |

test instance packages listed in Table 3 to generate test instances for MDASP. Most of them can be found on the website of Wei and Wenbin (2019), while the remaining ones were directly taken from the corresponding sources.

As mentioned above, except for the two instances from the competition, all other instances originate from 2D-KP or 2CSP. Therefore, in many cases the sum of the areas over all rectangles approximately equals the area of the given container, because here the focus often is on the placement of the rectangles within the container. Hence, if we simply declare them to be MDASP instances, the number of \underline{v} -good assortments, for $\underline{v} = (1 - \varepsilon)v^*$ with $\varepsilon = 0.05$, and the maximum diversities would often be rather small. On the other hand, if we consider an instance with a set of rectangles \mathcal{R} for which the sum of the areas is much bigger than the area of the container, $\mathcal{F}_{\underline{v}}$ may

contain many assortments having Hamming distance 1. Thus, we decided to modify the instances in order to ensure that $\mathcal{F}_{\underline{v}}$ has suitable size by scaling down the container and to thereby guarantee that at least one rectangle has to be contained in at least two assortments of each feasible selection.

Lemma 8 *Let \mathcal{I} be an instance of MDASP. Furthermore, let the total area of the given rectangles in \mathcal{R} be $A_{\mathcal{R}} := \sum_{R_i \in \mathcal{R}} w_i h_i$. If*

$$m\underline{v} > A_{\mathcal{R}}$$

holds, then at least one rectangle is contained in at least two assortments of each feasible selection.

Proof Assume there exist m feasible assortments $\mathcal{A}_1, \dots, \mathcal{A}_m$ such that the intersection of each pair of differing assortments is empty, i.e., $\mathcal{A}_i \cap \mathcal{A}_j = \emptyset$ for all $i, j \in \{1, \dots, m\}$ with $i \neq j$. Further, let $\mathcal{A}_U := \bigcup_{i \in \{1, \dots, m\}} \mathcal{A}_i$ be the union of the considered assortments. Since all assortments are feasible, we have $v(\mathcal{A}_i) \geq \underline{v}$ for all $i \in \{1, \dots, m\}$. Then it follows that

$$A_{\mathcal{R}} = \sum_{R_i \in \mathcal{R}} w_i h_i \geq \sum_{R_i \in \mathcal{A}_U} w_i h_i = \sum_{i=1}^m v(\mathcal{A}_i) \geq m\underline{v},$$

which is a contradiction. □

However, as we assume $\underline{v} = (1 - \varepsilon)v^*$ in the following and do not want to rely on determining v^* for every instance, we consider the area of the container $v(\mathcal{C})$ instead. Hence, the goal is to scale the container such that

$$m \cdot v(\mathcal{C}) > A_{\mathcal{R}}.$$

On the other hand, we additionally want to avoid instances with $A_{\mathcal{R}} \leq v(\mathcal{C})$ since this could imply that $\mathcal{F}_{\underline{v}}$ consists of only few assortments for a small ε . Therefore, we additionally request that

$$v(\mathcal{C}) < A_{\mathcal{R}}.$$

Summing up, our goal is to scale the containers such that

$$1 < p := \frac{m \cdot v(\mathcal{C})}{A_{\mathcal{R}}} < m.$$

Our scaling procedure works as follows: If $p \notin (1, m)$, let $s_r := \frac{h}{w}$ be the side-ratio of \mathcal{C} , and let $w_{max} := \max_{R_i \in \mathcal{R}} \{w_i\}$ and $h_{max} := \max_{R_i \in \mathcal{R}} \{h_i\}$ be the maximal width and height among all rectangles in \mathcal{R} . Then, we determine the minimum value

of $p \in \mathbb{N}$ with $p \geq 2$ such that \mathcal{C} is scaled, the side-ratio s_r is preserved, and each rectangle of \mathcal{R} still fits into the resulting container. This can be done by determining

$$p_{min} = \left\lceil \max \left(2, \frac{m \cdot s_r \cdot w_{max}^2}{A_{\mathcal{R}}}, \frac{m \cdot h_{max}^2}{s_r \cdot A_{\mathcal{R}}} \right) \right\rceil.$$

The formula can be derived from $p = \frac{m \cdot w \cdot h}{A_{\mathcal{R}}}$, $s_r \cdot w = h$, and the fact that the rectangles with maximum width and height still have to fit into the container.

If $p_{min} \in \{2, \dots, m-1\}$, we determine the corresponding integral width and height

$$w_{new} = \left\lceil \sqrt{\frac{p_{min} \cdot A_{\mathcal{R}}}{m \cdot s_r}} \right\rceil \text{ and } h_{new} = \left\lceil \sqrt{\frac{p_{min} \cdot s_r \cdot A_{\mathcal{R}}}{m}} \right\rceil,$$

and scale \mathcal{C} accordingly. Otherwise, we use the original container. Note that if this procedure led to a container with a greater area than the original one, we eventually decided to leave \mathcal{C} unchanged. This applies to eleven of the instances from the packages *2csp*, *OKP*, and *others*. We decided to do that in order to preserve instance specific properties and to allow only assortments that would have been feasible w.r.t. the original container.

Note that if a certain instance is not equipped with a container, we proceed analogously by setting s_r to the average side-ratio of the rectangles in \mathcal{R} , which was done for the instances of the *area* package. Finally, we removed instances that occurred twice after the scaling procedure and ended up with a test set consisting of 1,199 instances. All scaled instances that were used for evaluating the presented solution approaches are available as csv files at <https://cloud.zib.de/s/P3FBm9Wbn499LHY>.

7.3 Evaluating solution approaches for 2D-KP

In the remainder of this manuscript, we use the following abbreviations for 2D-KP algorithms: Concerning the heuristics we use *LWF*, *GRASP*, and *IBHP* for the Least-Waste-First heuristic of Wei et al. (2009), the GRASP algorithm of Álvarez Valdés et al. (2005), and the IBHP heuristic of Shiangjen et al. (2018), respectively. Regarding MIP formulations, by *HC95* we refer to the IP model of Hadjiconstantinou and Christofides (1995), *BKRS09* abbreviates the IP model of Belov et al. (2009), and *EP09* corresponds to the MIP formulation of Egeblad and Pisinger (2009). Recall that we used a time limit of 3,600 seconds for any computation, so it may happen that the heuristics lead to better results than the exact approaches.

7.3.1 Evaluation with respect to the best generated assortment

First of all, we compare the above mentioned 2D-KP algorithms w.r.t. the value of their best generated assortment for each instance in the test set. The exact results can be found as a csv file at <https://cloud.zib.de/s/bn8bd7Wfwj5KgT9>. In Table 4, we present summarized results for the different instance packages. Note that the number of instances on which the different approaches obtained the best assortment do not

have to sum up to 1,199 as we counted instances on which two or more algorithms achieved the best solution multiple times. Additionally, one should keep in mind that we compare three exact approaches and three heuristics.

The IBHP heuristic outperformed all other approaches w.r.t. the number of instances on which it determined an assortment with biggest value. This is the case for 1,018 of 1,199 instances, i.e., on 84.9% of the test set. The second-best approach in this context is the LWF heuristic with 310 of the 1,199 instances (25.9%), followed by approach EP09 (17.1%), and the GRASP heuristic (16.7%). Thus, when comparing all instances at once, it seems that IBHP is best suited for obtaining the best generated assortment. However, if we look at the different instance packages in more detail, we can observe that while on packages with $|\mathcal{R}|_{max} > 197$ the best generated assortments were indeed obtained by IBHP, EP09 delivered the best results on packages with $|\mathcal{R}|_{max} \leq 22$. This meets the expectation of the observer as the exact approaches are likely to perform worse with a growing number of rectangles.

Furthermore, the reasons for the results of the heuristic approaches may be due to their underlying basic ideas. GRASP relies on the idea of improving an initially generated assortment by randomly exchanging and moving the contained rectangles. On the other hand, IBHP and LWF apply sophisticated placing procedures where the rectangles are scored and placed in a manner such that wasted space is avoided if possible. However, the placing procedure in IBHP works faster than the evaluation within LWF, because here every rectangle is scored based on the question if it could cause wasted space in the next step. Thus, LWF spends more computing time when determining the scores for the rectangles, which IBHP can use to generate a greater variety of assortments.

If we compare the MIP approaches with each other, we observe that EP09 leads to the best results, although HC95 solved bigger instances in terms of the cardinality of \mathcal{R} . This behavior might be explained by the ideas behind the MIP models. HC95 relies on binary variables indicating whether a certain position inside the container is covered by a placed rectangle or not. This can be advantageous for instances with a small container. In contrast, EP09 and BKRS09 make use of the relative positions of the rectangles in their models. In our experiments the constraints modelling the relative positions in EP09 seem to be more effective than representing the relative positions by intersections in the projections onto the axes, which is the idea behind BKRS09.

7.3.2 Evaluation with respect to the number of generated assortments

Next, we compare the three heuristic approaches for 2D-KP w.r.t. the number of assortments that were generated. We do this, since in the first stage of the generic two-stage heuristic, the space of feasible \underline{v} -good assortments $\mathcal{F}_{\underline{v}}$ has to be sampled and therefore the number of generated assortments is an important factor. Note that we do not consider the MIP approaches in this context, as they did not generate many feasible solutions during the solving process, even for small instances. The total number of generated assortments and the number of \underline{v} -good assortments, for $\underline{v} = (1 - \varepsilon)v^*$ with $\varepsilon = 0.05$, can be found as a csv file at <https://cloud.zib.de/s/bn8bd7Wfwj5KgT9>. Note that for the number of \underline{v} -good solutions we only counted

Table 4 Number of instances on which the algorithms obtained an assortment with biggest value

| Package | $ \mathcal{R} _{min}$ | $ \mathcal{R} _{max}$ | # Instances | HC95 | BKRS09 | EP09 | LWF | GRASP | IBHP |
|---------------|-----------------------|-----------------------|-------------|------|--------|-----------|-----------|----------|-------------|
| <i>2dsp</i> | 20 | 50 | 60 | 0 | 0 | 17 | 0 | 12 | 47 |
| <i>AH</i> | 1000 | 1000 | 360 | 0 | 0 | 0 | 95 | 1 | 264 |
| <i>area</i> | 33 | 500 | 6 | 0 | 0 | 0 | 1 | 0 | 6 |
| <i>beng</i> | 20 | 200 | 10 | 4 | 1 | 3 | 9 | 9 | 9 |
| <i>Burke</i> | 10 | 3152 | 13 | 2 | 1 | 2 | 8 | 1 | 12 |
| <i>bwmv</i> | 20 | 100 | 500 | 39 | 12 | 112 | 93 | 96 | 463 |
| <i>C</i> | 16 | 197 | 21 | 9 | 6 | 9 | 21 | 18 | 20 |
| <i>cgcut</i> | 16 | 62 | 3 | 1 | 1 | 1 | 0 | 1 | 3 |
| <i>CX</i> | 50 | 15,000 | 7 | 0 | 0 | 1 | 6 | 6 | 7 |
| <i>gcut</i> | 10 | 50 | 13 | 0 | 0 | 5 | 1 | 4 | 13 |
| <i>HC</i> | 7 | 21 | 6 | 6 | 4 | 6 | 3 | 5 | 4 |
| <i>leung</i> | 10 | 50 | 10 | 3 | 3 | 8 | 5 | 8 | 7 |
| <i>N</i> | 17 | 197 | 35 | 0 | 0 | 9 | 12 | 5 | 29 |
| <i>ngcut</i> | 7 | 22 | 12 | 10 | 9 | 12 | 7 | 11 | 9 |
| <i>Nice</i> | 25 | 1000 | 6 | 0 | 0 | 0 | 3 | 0 | 3 |
| <i>Nice36</i> | 100 | 5000 | 33 | 0 | 0 | 0 | 2 | 0 | 31 |
| <i>OKP</i> | 30 | 97 | 5 | 0 | 0 | 3 | 0 | 5 | 0 |
| <i>others</i> | 40 | 200 | 4 | 1 | 0 | 1 | 1 | 2 | 4 |
| <i>Path</i> | 25 | 1000 | 6 | 0 | 0 | 1 | 1 | 0 | 6 |
| <i>Path36</i> | 25 | 5000 | 33 | 0 | 0 | 0 | 16 | 0 | 33 |
| <i>PB</i> | 10 | 20 | 5 | 1 | 3 | 5 | 2 | 5 | 4 |
| <i>T</i> | 17 | 199 | 35 | 0 | 1 | 10 | 12 | 2 | 28 |
| <i>ZDF</i> | 580 | 75,032 | 16 | 0 | 0 | 0 | 12 | 9 | 16 |
| Σ | | | 1199 | 76 | 41 | 205 | 310 | 200 | 1018 |

undominated assortments, i.e., assortments which were not contained in any other. For the total number of assortments, we did not remove the dominated ones, as their number was too big to complete the removing process within one week of computation time per instance.

First, we compare the different algorithms w.r.t. the total number of generated assortments. In this case, IBHP obtained the most solutions on 921 of the 1,199 instances, i.e., on 76.8%, while the GRASP algorithm obtained the most on the remaining 278 ones (23.2%), see Table 5. If we take a closer look on the properties of the different instances, we can observe that GRASP obtained the most assortments on packages with $|\mathcal{R}|_{\max} \leq 197$ and *Burke*.

Next, if we consider the subset of \underline{v} -good assortments and remove the dominated ones from it, which we were able to achieve within a week for each of the instances, the described situation intensifies, see Table 5. In this case, IBHP obtained the most assortments on 1,003 of the 1,199 instances (83.7%). The biggest instance on which GRASP delivers results superior to IBHP consists of 500 rectangles and the percentage of instances where it performs best decreases to 19.4%. Interestingly, LWF was able to generate the biggest set of assortments on 11 instances.

This behavior can again be explained by the subroutines on which the heuristics rely. As the scoring procedure within LWF consumes much time, IBHP and GRASP produce a greater variety of assortments. Further, recall that GRASP did not perform as good as IBHP in our evaluation w.r.t. the value of the best generated assortment, see Sect. 7.3.1. Consequently, it obtains a smaller number of generated assortments in total as its search is limited to a smaller solution subspace. This is in line with the fact that the number of instances on which GRASP obtains the most assortments decreases when considering only \underline{v} -good assortments. Recall that we consider $\underline{v} = (1 - \varepsilon)v^*$, where v^* is the best solution value found by any of the approaches, which gives IBHP an advantage. The former observations may be explained by the randomness regarding its improvement procedure, which can, in contrast to a more guided approach like IBHP, be a drawback when considering instances with many rectangles.

7.4 Evaluation with respect to diversity

Based on the results from the two previous subsections, we instantiated the generic two-stage heuristic (*2SH*) with IBHP and the random exchange heuristic presented in Sect. 6 and use MIP formulation EP09 for the instantiation of the MIQP model.

Next, we are going to compare the results of the solution approaches, discussed in Sects. 5 and 6. Therefore, we generated instances of MDASP from the scaled 2D-KP instances by setting $m = 6$, $\delta \in \{\delta_{\min}, \delta_{\text{avg}}\}$, and $\underline{v} = (1 - \varepsilon)v^*$ with $\varepsilon = 0.05$, i.e., we created two test instances for each container and set of rectangles. Hence, we ran every solution approach on 2,398 instances of MDASP. Recall that we always consider v^* to be equal to the value of the best generated assortment of the corresponding 2D-KP instance, see Sect. 7.3.1. Thus, we ran the MIQP formulation, the corresponding Benders decomposition approach (*BD*), and the two-stage heuristic for δ_{\min} as well

Table 5 Number of instances on which the heuristics generated the most assortments in total and the most undominated \underline{v} -good assortments, for $\underline{v} = (1 - \epsilon)v^*$ with $\epsilon = 0.05$

| Package | $ \mathcal{R} _{min}$ | $ \mathcal{R} _{max}$ | # Instances | total | | \underline{v} -good | | IBHP | GRASP | IBHP |
|---------------|-----------------------|-----------------------|-------------|-------|-------|-----------------------|-------|------|-------|-------|
| | | | | LWF | GRASP | LWF | GRASP | | | |
| <i>2dexp</i> | 20 | 50 | 60 | 0 | 14 | 0 | 0 | 46 | 4 | 56 |
| <i>AH</i> | 1000 | 1000 | 360 | 0 | 0 | 0 | 0 | 360 | 0 | 360 |
| <i>area</i> | 33 | 500 | 6 | 0 | 0 | 0 | 0 | 6 | 0 | 6 |
| <i>beng</i> | 20 | 200 | 10 | 0 | 2 | 0 | 0 | 8 | 1 | 9 |
| <i>Burke</i> | 10 | 3152 | 13 | 0 | 7 | 0 | 1 | 6 | 5 | 9 |
| <i>bwmv</i> | 20 | 100 | 500 | 0 | 170 | 0 | 0 | 330 | 157 | 360 |
| <i>C</i> | 16 | 197 | 21 | 0 | 9 | 0 | 0 | 12 | 9 | 12 |
| <i>cgcut</i> | 16 | 62 | 3 | 0 | 2 | 0 | 0 | 1 | 2 | 1 |
| <i>CX</i> | 50 | 15,000 | 7 | 0 | 1 | 0 | 1 | 6 | 1 | 7 |
| <i>gcut</i> | 10 | 50 | 13 | 0 | 5 | 0 | 0 | 8 | 3 | 13 |
| <i>HC</i> | 7 | 21 | 6 | 0 | 6 | 0 | 1 | 0 | 6 | 3 |
| <i>leung</i> | 10 | 50 | 10 | 0 | 8 | 0 | 2 | 2 | 9 | 2 |
| <i>N</i> | 17 | 197 | 35 | 0 | 12 | 0 | 23 | 23 | 13 | 32 |
| <i>ngcut</i> | 7 | 22 | 12 | 0 | 12 | 0 | 0 | 0 | 12 | 7 |
| <i>Nice</i> | 25 | 1000 | 6 | 0 | 1 | 0 | 5 | 5 | 0 | 6 |
| <i>Nice36</i> | 100 | 5000 | 33 | 0 | 0 | 0 | 33 | 33 | 0 | 33 |
| <i>OKP</i> | 30 | 97 | 5 | 0 | 5 | 0 | 0 | 0 | 5 | 0 |
| <i>others</i> | 40 | 200 | 4 | 0 | 2 | 0 | 2 | 2 | 2 | 2 |
| <i>Path</i> | 25 | 1000 | 6 | 0 | 1 | 0 | 5 | 5 | 1 | 5 |
| <i>Path36</i> | 25 | 5000 | 33 | 0 | 1 | 0 | 32 | 32 | 0 | 33 |
| <i>PB</i> | 10 | 20 | 5 | 0 | 5 | 0 | 0 | 0 | 5 | 2 |
| <i>T</i> | 17 | 199 | 35 | 0 | 14 | 0 | 21 | 21 | 7 | 29 |
| <i>ZDF</i> | 580 | 75,032 | 16 | 0 | 1 | 0 | 15 | 15 | 0 | 16 |
| Σ | | | 1199 | 0 | 278 | 0 | 921 | 921 | 232 | 41003 |

Table 6 Number of instances on which the solution approaches obtained the best diversity w.r.t. δ_{min} and δ_{avg}

| Package | $ \mathcal{R} _{min}$ | $ \mathcal{R} _{max}$ | # Instances | δ_{min} | | | δ_{avg} | | |
|---------------|-----------------------|-----------------------|-------------|----------------|----------|-------------|----------------|-----------|--------------|
| | | | | MIQP | BD | 2SH | MIQP | BD | 2SH |
| <i>2dcs</i> | 20 | 50 | 60 | 0 | 1 | 59 | 0 | 2 | 58 |
| <i>AH</i> | 1000 | 1000 | 360 | 0 | 0 | 360 | 0 | 0 | 360 |
| <i>area</i> | 33 | 500 | 6 | 0 | 0 | 6 | 0 | 0 | 6 |
| <i>beng</i> | 20 | 200 | 10 | 0 | 1 | 9 | 0 | 2 | 8 |
| <i>Burke</i> | 10 | 3152 | 13 | 1 | 1 | 10 | 1 | 1 | 10 |
| <i>bwmv</i> | 20 | 100 | 500 | 3 | 24 | 471 | 2 | 49 | 446 |
| <i>C</i> | 16 | 197 | 21 | 0 | 3 | 19 | 0 | 3 | 18 |
| <i>cgc</i> | 16 | 62 | 3 | 0 | 0 | 3 | 0 | 0 | 3 |
| <i>CX</i> | 50 | 15,000 | 7 | 1 | 0 | 6 | 1 | 0 | 6 |
| <i>gcut</i> | 10 | 50 | 13 | 1 | 1 | 11 | 1 | 3 | 13 |
| <i>HC</i> | 7 | 21 | 6 | 1 | 4 | 0 | 2 | 6 | 0 |
| <i>leung</i> | 10 | 50 | 10 | 1 | 3 | 8 | 1 | 3 | 7 |
| <i>N</i> | 17 | 197 | 35 | 0 | 6 | 29 | 0 | 9 | 26 |
| <i>ngcut</i> | 7 | 22 | 12 | 4 | 8 | 2 | 7 | 12 | 2 |
| <i>Nice</i> | 25 | 1000 | 6 | 0 | 0 | 6 | 0 | 0 | 6 |
| <i>Nice36</i> | 100 | 5000 | 33 | 0 | 0 | 33 | 0 | 0 | 33 |
| <i>OKP</i> | 30 | 97 | 5 | 0 | 0 | 5 | 0 | 0 | 5 |
| <i>others</i> | 40 | 200 | 4 | 0 | 0 | 4 | 0 | 0 | 4 |
| <i>Path</i> | 25 | 1000 | 6 | 0 | 0 | 6 | 0 | 1 | 5 |
| <i>Path36</i> | 25 | 5000 | 33 | 0 | 0 | 33 | 0 | 1 | 32 |
| <i>PB</i> | 10 | 20 | 5 | 1 | 3 | 1 | 2 | 4 | 1 |
| <i>T</i> | 17 | 199 | 35 | 2 | 5 | 29 | 1 | 7 | 27 |
| <i>ZDF</i> | 580 | 75,032 | 16 | 0 | 0 | 14 | 0 | 0 | 16 |
| Σ | | | 1199 | 15 | 60 | 1124 | 18 | 103 | 1,092 |

as for δ_{avg} . The detailed results we obtained can be found as a csv file at <https://cloud.zib.de/s/bn8bd7Wfwj5KgT9>.

7.4.1 Evaluation with respect to δ_{min}

First, we evaluate the results w.r.t. δ_{min} . Here, the heuristic delivered the best results on 1,124 of the 1,199 instances, i.e., on 93.7% of the instances and thus was the best performing approach among all considered algorithms, see Table 6. The second-best performing one in this context was the Benders approach, solving 60 instances best, i.e., 5.0%. However, the MIQP formulation was able to solve ten instances to optimality, while the heuristic obtained only four optimal selections, see Table 7.

When we consider the maximum number of rectangles contained in the instances and investigate the single packages in more detail, one can observe that the heuristic performs well on any type of test instance, and outperforms the other approaches on

Table 7 Minimum and maximum cardinality of \mathcal{R} of the instances that are solved best by the approaches w.r.t. δ_{min} and δ_{avg}

| | δ_{min} | | | δ_{avg} | | |
|-----------------------|----------------|----|--------|----------------|-----|--------|
| | MIQP | BD | 2SH | MIQP | BD | 2SH |
| $ \mathcal{R} _{max}$ | 50 | 29 | 25,032 | 50 | 120 | 75,032 |
| $ \mathcal{R} _{min}$ | 10 | 10 | 10 | 7 | 7 | 7 |
| # Optimal | 10 | 0 | 4 | 6 | 15 | 0 |

20 of the 23 packages. However, it performs less well on packages where the instances consist only of few rectangles, see for example *HC*, *ngcut*, and *PB*. Meanwhile, the Benders approach BD is especially strong when solving instances of this type and led to the best results on small instances, even if the biggest instance on which it could obtain the best diversity value consisted of 29 rectangles only. Furthermore, it was able to generate the most diverse selections on all instance packages with 22 rectangles or less. Finally, the biggest instance, w.r.t. the number of contained rectangles, for which the most diverse solution was found by the MIQP formulation, consisted of 50 rectangles.

Possible reasons for these results are on the one hand that the exact approaches, i.e., MIQP and BD, are likely to perform worse with an increasing number of rectangles. However, the exact approaches benefit from their ability to choose “less dense” assortments over “dense” ones. By this we mean that it may be advantageous to remove rectangles from an assortment as long as it stays \underline{v} -good in order to increase the diversity of the selection. The two-stage heuristic relying on IBHP does not consider this as it aims at filling up empty space in the constructed assortments by adding rectangles as long as possible. Nevertheless, this ability seems to be advantageous for instances with few rectangles and in particular for those, which were originally designed in a way that nearly all rectangles can be placed.

7.4.2 Evaluation with respect to δ_{avg}

For δ_{avg} , we make similar observations. Here, the heuristic obtained the best result on 1,092 instances, while the Benders approach could solve slightly more instances better than it did in the case of δ_{min} , i.e., 8.6%. Thus, this approach seems to perform better in the Average-Distance-Diversity case and its dominance on small instances w.r.t. the cardinality of \mathcal{R} intensifies. Meanwhile, the MIQP formulation could solve nearly the same amount of instances, see Table 6. Additionally, the Benders approach is now able to obtain the best result on instances with a size of up to 120 rectangles.

The obtained results can be explained analogously to the δ_{min} case, with the additional comment that the Benders approach performed better when considering δ_{avg} instead of δ_{min} . This may be due to the corresponding objective functions. When we are considering δ_{min} , changing one assortment in a selection does often not affect the overall diversity of the selection since the objective function aims at maximizing the minimum distance between the assortments. However, for δ_{avg} , the goal is to maximize the sum of the distances between the assortments, so an exchange of assortments

in a selection has nearly always a direct influence on the objective function. Hence, the search, which relies on a branch-and-bound tree, is more guided in the latter case.

Thus, we conclude again that the two-stage algorithm using IBHP and the random exchange heuristic is the best approach for deriving good solutions for MDASP instances, but when considering the Average-Distance-Diversity and an instance consisting of only a few rectangles, i.e., of less than 22, then the Benders approach is the solution approach of choice.

7.5 Evaluation of upper bounds

Finally, we look at the bounds obtained by MIQPs UB_{min} and UB_{avg} from Sect. 4. Recall that UB_{min} is only valid for δ_{min} , while UB_{avg} determines a valid bound for both diversity functions. It is important to note that any upper bound derived during the solving process is also valid. The results can be found as a csv file at <https://cloud.zib.de/s/bn8bd7Wfwj5KgT9>.

7.5.1 Evaluation with respect to δ_{min}

We start our evaluation with a comparison of both bounds on the instances w.r.t. δ_{min} . The minimum distance between any bound and the best result of any solution approach to MDASP is in both cases equal to 0, due to instance *ngcut01* since the maximum diversity that could be obtained coincides with the value of the bounds. Furthermore, the maximum distance between the bounds and best obtained diversity is 1, as for the biggest instances w.r.t. the number of contained rectangles, only selections with diversity equal to 0 are obtained, while the bounds are equal to 1.

Concerning the average distance between the bounds and the best obtained diversity, we can conclude that both bounds are nearly equally strong, with a slight advantage for UB_{avg} . But UB_{min} leads to better results on instances with less than 500 rectangles, making it better suited for smaller instances, while UB_{avg} performs best on instances with $|\mathcal{R}| \geq 500$, see Table 8. Furthermore, UB_{min} was the best bound for 552 instances, and UB_{avg} for 730 instances. On how many instances each bound performed best can be found in Table 9.

Note that, due to Lemma 7, UB_{min} should take on smaller values than UB_{avg} but since not all bounds could obtain their optimal value, we end up with UB_{avg} determining better bounds than UB_{min} on more than half of the instances, see Table 9. In fact, the MIQP formulations of UB_{min} and UB_{avg} were only able to obtain optimal solutions on 29 and 56 instances, respectively.

One possible explanation why UB_{avg} obtained better results than UB_{min} could again be due to the objective functions of the underlying MIQPs. While UB_{avg} minimizes a sum of variables, UB_{min} aims at minimizing a single value which often works worse in practice.

7.5.2 Evaluation with respect to δ_{avg}

Finally, we present the results obtained by UB_{avg} for the instances w.r.t. δ_{avg} . The minimum distance between UB_{avg} and the value of a most diverse selection is again

Table 8 Average distance between the bounds and the best obtained diversity. δ^* signifies the value of the most diverse selection that could be obtained by any solution approach to MDASP

| Package | $ \mathcal{R} _{min}$ | $ \mathcal{R} _{max}$ | # Instances | $\frac{\delta_{min}}{UB_{min} - \delta^*_{min}}$ | $\frac{UB_{avg} - \delta^*_{min}}{UB_{avg} - \delta^*_{min}}$ | $\frac{\delta_{avg}}{UB_{avg} - \delta^*_{avg}}$ |
|---------------|-----------------------|-----------------------|-------------|--|---|--|
| <i>2dscsp</i> | 20 | 50 | 60 | 0.174 | 0.188 | 0.127 |
| <i>AH</i> | 1000 | 1000 | 360 | 0.307 | 0.274 | 0.225 |
| <i>area</i> | 33 | 500 | 6 | 0.256 | 0.253 | 0.190 |
| <i>beng</i> | 20 | 200 | 10 | 0.224 | 0.232 | 0.174 |
| <i>Burke</i> | 10 | 3152 | 13 | 0.310 | 0.303 | 0.261 |
| <i>bwmv</i> | 20 | 100 | 500 | 0.265 | 0.267 | 0.227 |
| <i>C</i> | 16 | 197 | 21 | 0.215 | 0.233 | 0.164 |
| <i>cgcut</i> | 16 | 62 | 3 | 0.457 | 0.525 | 0.472 |
| <i>CX</i> | 50 | 15,000 | 7 | 0.422 | 0.419 | 0.371 |
| <i>gcut</i> | 10 | 50 | 13 | 0.332 | 0.362 | 0.289 |
| <i>HC</i> | 7 | 21 | 6 | 0.261 | 0.336 | 0.210 |
| <i>leung</i> | 10 | 50 | 10 | 0.231 | 0.287 | 0.197 |
| <i>N</i> | 17 | 197 | 35 | 0.272 | 0.311 | 0.229 |
| <i>ngcut</i> | 7 | 22 | 12 | 0.112 | 0.154 | 0.064 |
| <i>Nice</i> | 25 | 1000 | 6 | 0.165 | 0.187 | 0.155 |
| <i>Nice36</i> | 100 | 5000 | 33 | 0.446 | 0.439 | 0.396 |
| <i>OKP</i> | 30 | 97 | 5 | 0.687 | 0.689 | 0.578 |
| <i>others</i> | 40 | 200 | 4 | 0.180 | 0.201 | 0.115 |
| <i>Path</i> | 25 | 1000 | 6 | 0.254 | 0.240 | 0.200 |
| <i>Path36</i> | 25 | 5000 | 33 | 0.365 | 0.366 | 0.328 |
| <i>PB</i> | 10 | 20 | 5 | 0.256 | 0.367 | 0.306 |
| <i>T</i> | 17 | 199 | 35 | 0.281 | 0.308 | 0.255 |
| <i>ZDF</i> | 580 | 75,032 | 16 | 0.481 | 0.481 | 0.415 |
| \emptyset | | | 1199 | 0.285 | 0.281 | 0.232 |

Table 9 Number of instances per package on which the presented bounds performed best w.r.t. δ_{min}

| Package | $ \mathcal{R} _{min}$ | $ \mathcal{R} _{max}$ | # Instances | UB_{min} | UB_{avg} |
|---------------|-----------------------|-----------------------|-------------|------------|------------|
| <i>2dcsp</i> | 20 | 50 | 60 | 44 | 33 |
| <i>AH</i> | 1000 | 1000 | 360 | 5 | 315 |
| <i>area</i> | 33 | 500 | 6 | 3 | 3 |
| <i>beng</i> | 20 | 200 | 10 | 10 | 3 |
| <i>Burke</i> | 10 | 3152 | 13 | 3 | 10 |
| <i>bwmv</i> | 20 | 100 | 500 | 240 | 271 |
| <i>C</i> | 16 | 197 | 21 | 20 | 2 |
| <i>cgcut</i> | 16 | 62 | 3 | 3 | 0 |
| <i>CX</i> | 50 | 15,000 | 7 | 4 | 6 |
| <i>gcut</i> | 10 | 50 | 13 | 9 | 4 |
| <i>HC</i> | 7 | 21 | 6 | 6 | 0 |
| <i>leung</i> | 10 | 50 | 10 | 9 | 1 |
| <i>N</i> | 17 | 197 | 35 | 34 | 1 |
| <i>ngcut</i> | 7 | 22 | 12 | 12 | 1 |
| <i>Nice</i> | 25 | 1000 | 6 | 6 | 1 |
| <i>Nice36</i> | 100 | 5000 | 33 | 17 | 23 |
| <i>OKP</i> | 30 | 97 | 5 | 5 | 3 |
| <i>others</i> | 40 | 200 | 4 | 4 | 2 |
| <i>Path</i> | 25 | 1000 | 6 | 1 | 5 |
| <i>Path36</i> | 25 | 5000 | 33 | 19 | 29 |
| <i>PB</i> | 10 | 20 | 5 | 5 | 0 |
| <i>T</i> | 17 | 199 | 35 | 30 | 5 |
| <i>ZDF</i> | 580 | 75,032 | 16 | 12 | 12 |
| \emptyset | | | 1199 | 552 | 730 |

equal to 0, due to instance *ngcut01*. We also obtain a maximum distance of 1 between UB_{avg} and the best diversity due to instance *zdf16*.

The average distances to the best obtained diversities generated by the solution approaches to MDASP are depicted in the right column of Table 8. As expected, the distances between UB_{avg} and the value of the most diverse selection are smaller than in the case of δ_{min} since the diversity of a selection w.r.t. δ_{avg} is bigger than or equal to its diversity w.r.t. δ_{min} , see Lemma 3. Furthermore, UB_{avg} seems to work best on packages with instances consisting of 1,000 rectangles or less. Nevertheless, it also obtains its biggest distance on a package with instances consisting of less than 100 rectangles, i.e., on package *OKP*. This suggests that either the solution approaches fail on determining selections with great diversity or that the relaxation of the conditions that the considered assortments have to fulfill is too weak. But this issue occurs for the bounds on δ_{min} as well. However, with an average distance of 0.232 to all considered instances, UB_{avg} achieves better results for δ_{avg} than any bound on δ_{min} .

8 Conclusion and outlook

In this article, we introduced the Maximum Diversity Assortment Selection Problem (MDASP), which is a novel generalization of the two-dimensional Knapsack Problem (2D-KP). First, we mathematically defined MDASP and introduced two diversity-functions for selections based on the Hamming distance. Afterwards, we presented an overview of the literature focusing on its two inherent subproblems 2D-KP and MDP. Next, we introduced two MIQPs that can be used to determine upper bounds on the diversity of MDASP instances. Based on them, we presented an exact MIQP formulation for MDASP, a Benders decomposition approach for it, as well as a generic two-stage heuristic. Furthermore, we compared different solution approaches for 2D-KP with respect to the best assortment value and the number of generated assortments. Finally, we investigated the presented solution approaches for MDASP with respect to the maximum diverse selections they determined.

As a main result, the generic two-stage heuristic instantiated with IBHP and the random exchange for MDP delivered the best results with respect to the diversity among the presented solution approaches for MDASP. However, the Benders approaches led to more diverse selections on instances consisting of only few rectangles, especially with respect to δ_{avg} . Again, it is important to note that it is an exact algorithm, in contrast to the heuristic.

There are many directions for further research concerning MDASP. First of all, we are currently working on a possible improvement of the Benders decomposition approach by using the algorithm presented by Fekete et al. (2007) for determining the maximality of an assortment. Additionally, it would be beneficial to experiment with other cuts than the no-good-cuts in order to improve the performance of the overall Benders approach. For example, one could try to use a variant of “Combinatorial Benders Cuts” as suggested by Côté et al. (2014). Additionally, it would be interesting to test a sequence-pair representation based formulation within the presented MIQP approaches and try to speed up the computations and improve the LP bound by that.

Furthermore, we are investigating the structures of the solutions created by the different heuristics. The idea here would be to combine them in order to further broaden the variety in the set of generated \underline{v} -good solutions. In this context, a modification of the shift strategy of IBHP, adapting the strength of the GRASP heuristic for small instances, would be of great advantage. Additionally, it would be interesting if a heuristic exchange of rectangles with similar widths and heights could be integrated into the current approaches in order to improve the diversity between the constructed assortments.

Acknowledgements The work for this article has been conducted in the Research Campus MODAL funded by the German Federal Ministry of Education and Research (BMBF) (Fund Numbers 05M14ZAM, 05M20ZBM).

Funding Open Access funding enabled and organized by Projekt DEAL.

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

- Achterberg T (2007) Constraint Integer Programming. Doctoral thesis, Technische Universität Berlin, Fakultät II - Mathematik und Naturwissenschaften, Berlin. <https://doi.org/10.14279/depositonce-1634>
- AdaCore: GNAT Reference Manual (2019). <https://www.adacore.com/documentation>
- AIMMS-MOPTA optimization modeling competition. ORMS Today (2019). <https://pubsonline.informs.org/doi/10.1287/orms.2019.05.13/full/>
- Álvarez Valdés R, Parreño F, Tamarit JM (2005) A GRASP algorithm for constrained two-dimensional non-guillotine cutting problems. *J Oper Res Soc* 56(4):414–425. <https://doi.org/10.1057/palgrave.jors.2601829>
- Álvarez Valdés R, Parreño F, Tamarit JM (2007) A tabu search algorithm for a two-dimensional non-guillotine cutting problem. *Eur J Oper Res* 183(3):1167–1182. <https://doi.org/10.1016/j.ejor.2005.11.068>
- Babu AR, Babu NR (1999) Effective nesting of rectangular parts in multiple rectangular sheets using genetic and heuristic algorithms. *Int J Prod Res* 37(7):1625–1643. <https://doi.org/10.1080/002075499191166>
- Beasley JE (1985) Algorithms for unconstrained two-dimensional Guillotine cutting. *J Oper Res Soc* 36(4):297–306. <https://doi.org/10.1057/jors.1985.51>
- Beasley JE (1985) An exact two-dimensional non-guillotine cutting tree search procedure. *Oper Res* 33(1):49–64. <https://doi.org/10.1287/opre.33.1.49>
- Beasley JE (2004) A population heuristic for constrained two-dimensional non-guillotine cutting. *Eur J Oper Res* 156(3):601–627. [https://doi.org/10.1016/S0377-2217\(03\)00139-5](https://doi.org/10.1016/S0377-2217(03)00139-5)
- Belov G, Kartak V, Rohling H, Scheithauer G (2009) One-dimensional relaxations and LP bounds for orthogonal packing. *Int Trans Oper Res* 16(6):745–766. <https://doi.org/10.1111/j.1475-3995.2009.00713.x>
- Benders JF (2005) Partitioning procedures for solving mixed-variables programming problems. *Comput Manag Sci* 2(1):3–19. <https://doi.org/10.1007/s10287-004-0020-y>
- Bengtsson BE (1982) Packing rectangular pieces—a heuristic approach. *Comput J* 25(3):353–357. <https://doi.org/10.1093/comjnl/25.3.353>
- Berkey JO, Wang PY (1987) Two-dimensional finite bin-packing algorithms. *J Oper Res Soc* 38(5):423–429. <https://doi.org/10.1057/jors.1987.70>
- Bortfeldt A, Winter T (2009) A genetic algorithm for the two-dimensional knapsack problem with rectangular pieces. *Int Trans Oper Res* 16(6):685–713. <https://doi.org/10.1111/j.1475-3995.2009.00701.x>
- Bortfeldt A, Gehring H (2006) New large benchmark instances for the two-dimensional strip packing problem with rectangular pieces. In: Proceedings of the 39th annual Hawaii international conference on system sciences (HICSS'06), pp. 30b–30b. IEEE, Kauai, HI, USA. <https://doi.org/10.1109/HICSS.2006.360>. <http://ieeexplore.ieee.org/document/1579352/>
- Boschetti MA, Mingozzi A, Hadjiconstantinou E (2002) New upper bounds for the two-dimensional orthogonal non-guillotine cutting stock problem. *IMA J Manag Math* 13(2):95–119. <https://doi.org/10.1093/imaman/13.2.95>
- Bouaine A, Lebbar M, Ha MA (2018) Minimization of the wood wastes for an industry of furnishing: a two dimensional cutting stock problem. *Manag Prod Eng Rev*. <https://doi.org/10.24425/119524>
- Burke EK, Kendall G, Whitwell G (2004) A new placement heuristic for the orthogonal stock-cutting problem. *Oper Res* 52(4):655–671. <https://doi.org/10.1287/opre.1040.0109>
- Cheng CH, Feiring BR, Cheng TCE (1994) The cutting stock problem—a survey. *Int J Prod Econ* 36(3):291–305. [https://doi.org/10.1016/0925-5273\(94\)00045-X](https://doi.org/10.1016/0925-5273(94)00045-X)
- Christofides N, Whitlock C (1977) An algorithm for two-dimensional cutting problems. *Oper Res* 25(1):30–44. <https://doi.org/10.1287/opre.25.1.30>
- Clautiaux F, Carlier J, Moukrim A (2007) A new exact method for the two-dimensional orthogonal packing problem. *Eur J Oper Res* 183(3):1196–1211. <https://doi.org/10.1016/j.ejor.2005.12.048>

- Côté JF, DellAmico M, Iori M (2014) Combinatorial benders cuts for the strip packing problem. *Oper Res* 62(3):643–661. <https://doi.org/10.1287/opre.2013.1248>
- Crainic TG, Perboli G, Tadei R (2012) Recent advances in multi-dimensional packing problems. In: Volosencu C (Eds) *New technologies—trends, innovations and research*. InTech. <https://doi.org/10.5772/33302>. <http://www.intechopen.com/books/new-technologies-trends-innovations-and-research/recent-advances-in-multi-dimensional-packing-problems>
- de Armas J, Miranda G, León C (2012) Improving the efficiency of a best-first bottom-up approach for the Constrained 2d Cutting Problem. *Eur J Oper Res* 219(2):201–213. <https://doi.org/10.1016/j.ejor.2011.11.002>
- Dowland KA, Dowland WB (1992) Packing problems. *Eur J Oper Res* 56(1):2–14. [https://doi.org/10.1016/0377-2217\(92\)90288-K](https://doi.org/10.1016/0377-2217(92)90288-K)
- Duarte A, Martí R (2007) Tabu search and GRASP for the maximum diversity problem. *Eur J Oper Res* 178(1):71–84. <https://doi.org/10.1016/j.ejor.2006.01.021>
- Dyckhoff H (1990) A typology of cutting and packing problems. *Eur J Oper Res* 44(2):145–159. [https://doi.org/10.1016/0377-2217\(90\)90350-K](https://doi.org/10.1016/0377-2217(90)90350-K)
- Egeblad J, Pisinger D (2009) Heuristic approaches for the two- and three-dimensional knapsack packing problem. *Comput Oper Res* 36(4):1026–1049. <https://doi.org/10.1016/j.cor.2007.12.004>
- Fekete SP, Schepers J (2000) On more-dimensional packing III: exact algorithms
- Fekete SP, Schepers J (2004) A combinatorial characterization of higher-dimensional orthogonal packing. *Math Oper Res* 29(2):353–368. <https://doi.org/10.1287/moor.1030.0079>
- Fekete SP, Schepers J, van der Veen JC (2007) An exact algorithm for higher-dimensional orthogonal packing. *Oper Res* 55(3):569–587. <https://doi.org/10.1287/opre.1060.0369>
- Gallego M, Duarte A, Laguna M, Martí R (2009) Hybrid heuristics for the maximum diversity problem. *Comput Optim Appl* 44(3):411–426. <https://doi.org/10.1007/s10589-007-9161-6>
- Garey MR, Johnson DS (1979) *Computers and intractability: a guide to the theory of NP-completeness*. W. H. Freeman, San Francisco. <https://bohr.wlu.ca/hfan/cp412/references/ChapterOne.pdf>
- Geoffrion AM (1972) Generalized Benders decomposition. *J Optim Theory Appl* 10(4):237–260. <https://doi.org/10.1007/BF00934810>
- Ghosh JB (1996) Computational aspects of the maximum diversity problem. *Oper Res Lett* 19(4):175–181. [https://doi.org/10.1016/0167-6377\(96\)00025-9](https://doi.org/10.1016/0167-6377(96)00025-9)
- Gilmore PC, Gomory RE (1965) Multistage cutting stock problems of two and more dimensions. *Oper Res* 13(1):94–120. <https://doi.org/10.1287/opre.13.1.94>
- Gonçalves JF, Resende MGC (2006) A hybrid heuristic for the constrained two-dimensional non-guillotine orthogonal cutting problem. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.73.3368>
- Gonçalves JF (2007) A hybrid genetic algorithm-heuristic for a two-dimensional orthogonal packing problem. *Eur J Oper Res* 183(3):1212–1229. <https://doi.org/10.1016/j.ejor.2005.11.062>
- Gurobi optimization, LLC: Gurobi Optimizer Reference Manual (2019). <http://www.gurobi.com>
- Hadjiconstantinou E, Christofides N (1995) An exact algorithm for general, orthogonal, two-dimensional knapsack problems. *Eur J Oper Res* 83(1):39–56. [https://doi.org/10.1016/0377-2217\(93\)E0278-6](https://doi.org/10.1016/0377-2217(93)E0278-6)
- Hadjiconstantinou E, Iori M (2007) A hybrid genetic algorithm for the two-dimensional single large object placement problem. *Eur J Oper Res* 183(3):1150–1166. <https://doi.org/10.1016/j.ejor.2005.11.061>
- Haessler RW (1971) A heuristic programming solution to a nonlinear cutting stock problem. *Manag Sci* 17(12):793. <https://doi.org/10.1287/mnsc.17.12.B793>
- Hahn SG (1968) On the optimal cutting of defective sheets. *Oper Res* 16(6):1100–1114. <https://doi.org/10.1287/opre.16.6.1100>
- Hamming RW (1950) Error detecting and error correcting codes. *Bell Syst Tech J* 29(2):147–160. <https://doi.org/10.1002/j.1538-7305.1950.tb00463.x>
- Hatefi MA (2017) Developing column generation approach to solve the rectangular two-dimensional single knapsack problem. *Sci Iran* 24(6):3287–3296. <https://doi.org/10.24200/sci.2017.4401>
- He K, Huang W, Jin Y (2012) An efficient deterministic heuristic for two-dimensional rectangular packing. *Comput Oper Res* 39(7):1355–1363. <https://doi.org/10.1016/j.cor.2011.08.005>
- Hifi M (2001) Exact algorithms for large-scale unconstrained two and three staged cutting problems. *Comput Optim Appl* 18(1):63–88. <https://doi.org/10.1023/A:1008743711658>
- Hifi M, Zissimopoulos V (1997) Constrained two-dimensional cutting: an improvement of Christofides and Whitlocks exact algorithm. *J Oper Res Soc* 48(3):324–331. <https://doi.org/10.1057/palgrave.jors.2600364>

- Hinxman AI (1980) The trim-loss and assortment problems: a survey. *Eur J Oper Res* 5(1):8–18. [https://doi.org/10.1016/0377-2217\(80\)90068-5](https://doi.org/10.1016/0377-2217(80)90068-5)
- Hopper E (2000) Two-dimensional packing utilising evolutionary algorithms and other meta-heuristic methods. Ph.D. thesis, University of Wales, Cardiff. <http://vmk.ugatu.ac.ru/c%26p/article/hopper/PhDisser/part1.pdf>
- Hopper E, Turton BCH (2001) An empirical investigation of meta-heuristic and heuristic algorithms for a 2d packing problem. *Eur J Oper Res* 128(1):34–57. [https://doi.org/10.1016/S0377-2217\(99\)00357-4](https://doi.org/10.1016/S0377-2217(99)00357-4)
- Huang W, Chen D (2007) An efficient heuristic algorithm for rectangle-packing problem. *Simul Model Pract Theory* 15(10):1356–1365. <https://doi.org/10.1016/j.simpat.2007.09.004>
- Imahori S (2019) Cutting and Packing. Accessed 2019-11-13. <http://www-or.amp.i.kyoto-u.ac.jp/~imahori/packing/index.html>
- Iori M, de Lima VL, Martello S, Miyazawa FK, Monaci M (2020) Exact solution techniques for two-dimensional cutting and packing. *Eur J Oper Res*. <https://doi.org/10.1016/j.ejor.2020.06.050>
- Jakobs S (1996) On genetic algorithms for the packing of polygons. *Eur J Oper Res* 88(1):165–181
- Kantorovich LV (1960) Mathematical methods of organizing and planning production. *Manag Sci* 6(4):366–422. <https://doi.org/10.1287/mnsc.6.4.366>
- Kök AG, Fisher ML, Vaidyanathan R (2015) Assortment planning: Review of literature and industry practice. In: *Retail supply chain management*, pp. 175–236. Springer. https://doi.org/10.1007/978-1-4899-7562-1_8
- Kuo CC, Glover F, Dhir KS (1993) Analyzing and modeling the maximum diversity problem by zero-one programming. *Decis Sci* 24(6):1171–1185. <https://doi.org/10.1111/j.1540-5915.1993.tb00509.x>
- Lai KK, Chan JWM (1997) Developing a simulated annealing algorithm for the cutting stock problem. *Comput Ind Eng* 32(1):115–127. [https://doi.org/10.1016/S0360-8352\(96\)00205-7](https://doi.org/10.1016/S0360-8352(96)00205-7)
- Leung TW, Chan CK, Troutt MD (2003) Application of a mixed simulated annealing-genetic algorithm heuristic for the two-dimensional orthogonal packing problem. *Eur J Oper Res* 145(3):530–542. [https://doi.org/10.1016/S0377-2217\(02\)00218-7](https://doi.org/10.1016/S0377-2217(02)00218-7)
- Leung SCH, Zhang D, Zhou C, Wu T (2012) A hybrid simulated annealing metaheuristic algorithm for the two-dimensional knapsack packing problem. *Comput Oper Res* 39(1):64–73. <https://doi.org/10.1016/j.cor.2010.10.022>
- Lodi A, Martello S, Vigo D (1999) Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems. *INFORMS J Comput* 11(4):345–357. <https://doi.org/10.1287/ijoc.11.4.345>
- Lozano M, Molina D, García-Martínez C (2011) Iterated greedy for the maximum diversity problem. *Eur J Oper Res* 214(1):31–38. <https://doi.org/10.1016/j.ejor.2011.04.018>
- Martello S, Vigo D (1998) Exact solution of the two-dimensional finite bin packing problem. *Manag Sci* 44(3):388–399. <https://doi.org/10.1287/mnsc.44.3.388>
- Martí R, Gallego M, Duarte A (2010) A branch and bound algorithm for the maximum diversity problem. *Eur J Oper Res* 200(1):36–44. <https://doi.org/10.1016/j.ejor.2008.12.023>
- Martí R, Gallego M, Duarte A, Pardo EG (2013) Heuristics and metaheuristics for the maximum diversity problem. *J Heurist* 19(4):591–615. <https://doi.org/10.1007/s10732-011-9172-4>
- MirHassani SA, Jalaiean Bashirzadeh A (2015) A GRASP meta-heuristic for two-dimensional irregular cutting stock problem. *Int J Adv Manuf Technol* 81(1–4):455–464. <https://doi.org/10.1007/s00170-015-7107-1>
- Murata H, Fujiyoshi K, Nakatake S, Kajitani Y (2003) Rectangle-packing-based module placement. In: *The best of ICCAD*. Springer, pp 535–548. https://doi.org/10.1007/978-1-4615-0292-0_42
- Padberg M (2000) Packing small boxes into a big box. *Math Methods Oper Res* 52(1):1–21. <https://doi.org/10.1007/s001860000066>
- Perboli G, Crainic TG, Tadei R (2011) An efficient metaheuristic for multi-dimensional multi-container packing. In: 2011 IEEE international conference on automation science and engineering, pp. 563–568. IEEE, Trieste, Italy. <https://doi.org/10.1109/CASE.2011.6042476>. <http://ieeexplore.ieee.org/document/6042476/>
- Pinto E, Oliveira JF (2005) Algorithm based on graphs for the non-guillotinable two-dimensional packing problem. In: *Second ESICUP meeting*, Southampton
- Pisinger D (2007) Denser packings obtained in $O(n \log \log n)$ time. *INFORMS J Comput* 19(3):395–405. <https://doi.org/10.1287/ijoc.1060.0192>
- Problem description for the 11th AIMMS-MOPTA Competition (2019). https://coral.ise.lehigh.edu/~mopta2019/mopta2019/AIMMS_MOPTA_case_2019.pdf
- Python Software Foundation: Python 3.6 documentation (2020). <https://docs.python.org/3.6/>

- Ravi SS, Rosenkrantz DJ, Tayi GK (1994) Heuristic and special case algorithms for dispersion problems. *Oper Res* 42(2):299–310. <https://doi.org/10.1287/opre.42.2.299>
- Resende MGC, Martí R, Gallego M, Duarte A (2010) GRASP and path relinking for the max-min diversity problem. *Comput Oper Res* 37(3):498–508. <https://doi.org/10.1016/j.cor.2008.05.011>
- Sandoya F, Martínez-Gavara A, Aceves R, Duarte A, Martí R (2018) Diversity and equity models. In: R. Martí, P.M. Pardalos, M.G.C. Resende (eds.) *Handbook of heuristics*, pp. 979–998. Springer International Publishing. <https://www.springer.com/gp/book/9783319071237>
- Santos LF, Ribeiro MH, Plastino A, Martins SL (2005) A Hybrid GRASP with data mining for the maximum diversity problem. In: Hutchison D, Kanade T, Kittler J, Kleinberg JM, Mattern K, Mitchell JC, Naor M, Nierstrasz O, Pandu Rangan C, Steffen B, Sudan M, Terzopoulos D, Tygar D, Vardi MY, Weikum G, Blesa MJ, Blum C, Rolí A, Sampels M (Eds) *Hybrid metaheuristics*, vol. 3636, pp. 116–127. Springer Berlin Heidelberg, Berlin, Heidelberg. https://doi.org/10.1007/11546245_11. http://link.springer.com/10.1007/11546245_11
- Shiangjen K, Chaijaruwanch J, Srisujalertwaja W, Unachak P, Somhom S (2018) An iterative bidirectional heuristic placement algorithm for solving the two-dimensional knapsack packing problem. *Eng Optim* 50(2):347–365. <https://doi.org/10.1080/0305215X.2017.1315571>
- Silva GC, Ochi LS, Martins SL (2004) Experimental comparison of greedy randomized adaptive search procedures for the maximum diversity problem. In: Kanade T, Kittler J, Kleinberg JM, Mattern F, Mitchell JC, Nierstrasz O, Pandu Rangan C, Steffen B, Sudan M, Terzopoulos D, Tygar D, Vardi MY, Weikum G, Ribeiro CC, Martins SL (Eds.) *Experimental and efficient algorithms*, vol. 3059, pp 498–512. Springer, Berlin. https://doi.org/10.1007/978-3-540-24838-5_37
- Silva GC, de Andrade MRQ, Ochi LS, Martins SL, Plastino A (2007) New heuristics for the maximum diversity problem. *J Heuristics* 13(4):315–336. <https://doi.org/10.1007/s10732-007-9010-x>
- Wang S (2017) Solving rectangle packing problem based on heuristic dynamic decomposition algorithm. *DEStech Transactions on Engineering and Technology Research (eeta)*. <https://doi.org/10.12783/dtetr/eeta2017/7728>
- Wang PY (1983) Two algorithms for constrained two-dimensional cutting stock problems. *Oper Res* 31(3):573–586. <https://doi.org/10.1287/opre.31.3.573>
- Wang PY, Valenzuela CL (2001) Data set generation for rectangular placement problems. *Eur J Oper Res* 134(2):378–391. [https://doi.org/10.1016/S0377-2217\(00\)00263-0](https://doi.org/10.1016/S0377-2217(00)00263-0)
- Wäscher G, Haußner H, Schumann H (2007) An improved typology of cutting and packing problems. *Eur J Oper Res* 183(3):1109–1130. <https://doi.org/10.1016/j.ejor.2005.12.047>
- Wei L, Zhang D, Chen Q (2009) A least wasted first heuristic algorithm for the rectangular packing problem. *Comput Oper Res* 36(5):1608–1614. <https://doi.org/10.1016/j.cor.2008.03.004>
- Wei L, Wenbin Z (2019) Skyline heuristic for the 2d rectangular packing and strip packing problems. Accessed 2019-11-13. <https://www.computational-logistics.org/orlib/topic/2D%20Strip%20Packing/index.html#data>
- Wu YL, Chan CK (2005) On improved least flexibility first heuristics superior for packing and stock cutting problems. In: Hutchison D, Kanade T, Kittler J, Kleinberg JM, Mattern F, Mitchell JC, Naor M, Nierstrasz O, Pandu Rangan C, Steffen B, Sudan M, Terzopoulos D, Tygar D, Vardi MY, Weikum G, Lupanov OB, Kasim-Zade OM, Chaskin AV, Steinhöfel K (Eds) *Stochastic algorithms: foundations and applications*, vol. 3777, pp 70–81. Springer, Berlin. https://doi.org/10.1007/11571155_8
- Wu YL, Huang W, Lau SC, Wong CK, Young GH (2002) An effective quasi-human based heuristic for solving the rectangle packing problem. *Eur J Oper Res* 141(2):341–358. [https://doi.org/10.1016/S0377-2217\(02\)00129-7](https://doi.org/10.1016/S0377-2217(02)00129-7)