European Journal of Operational Research xxx (xxxx) xxx

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Invited Review

A review on discrete diversity and dispersion maximization from an OR perspective

Rafael Martí[a,*], Anna Martínez-Gavara[a], Sergio Pérez-Peló[b], Jesús Sánchez-Oro[b]

[a] Departament d'Estadística i Investigació Operativa, Universitat de València, Spain
[b] Departamento de Ciencias de la Computación, Universidad Rey Juan Carlos, Spain

ARTICLE INFO

Article history:
Received 9 March 2021
Accepted 24 July 2021
Available online xxx

Keywords:
Combinatorial optimization
Diversity
Dispersion
Mathematical models
Metaheuristics

ABSTRACT

The problem of maximizing diversity or dispersion deals with selecting a subset of elements from a given set in such a way that the distance among the selected elements is maximized. The definition of distance between elements is customized to specific applications, and the way that the overall diversity of the selected elements is computed results in different mathematical models. Maximizing diversity by means of combinatorial optimization models has gained prominence in Operations Research (OR) over the last two decades, and constitutes nowadays an important area. In this paper, we review the milestones in the development of this area, starting in the late eighties when the first models were proposed, and identify three periods of time. The critical analysis from an OR perspective of the previous developments, permits us to establish the most appropriate models, their connection with practical problems in terms of dispersion and representativeness, and the open problems that are still a challenge. We also revise and extend the library of benchmark instances that has been widely used in heuristic comparisons. Finally, we perform an empirical review and comparison of the best and more recently proposed procedures, to clearly identify the state-of-the art methods for the main diversity models.

1. Introduction

Maximum diversity problems arise in many practical settings from facility location to social network analysis, and constitute an important class of NP-hard problems in combinatorial optimization. They were first approached from an Operations Research perspective in 1988 by Kuby (1988) and presented in 1993 in the annual meeting of the Decision Science Institute, where Kuo, Glover, and Dhir, proposed integer programming models (Dhir, Glover, & Kuo, 1993; Kuo, Glover, & Dhir, 1993). There has been a growing interest in these problems in the last 30 years, and different mathematical programming models, and their corresponding solving methods, have been proposed to capture the notion of diversity. They basically consist in selecting a subset of elements of a given set, in such a way that a distance measure is maximized, and differ among them in the way that the overall diversity of the selected elements is computed. In its graph version, the most popular dispersion model, the Maximum Diversity Problem (MDP), is defined as follows. Given the complete graph $G = (V, E)$ with edge distances d_{ij} for every pair $i, j \in V$, and an integer m, compute a subset M of V, such that $|M| = m$ and $\sum_{i,j \in M} d_{ij}$ is as large as possible.

The study of diversity models, also called dispersion, has achieved a level of maturity, and still has a huge potential, which makes it especially adequate for a review paper like this one. In our opinion, we are witnessing the typical scenario in science, in which a sub-field of research detaches from the main field and creates its own body of knowledge. Diversity problems may be considered, in a certain way, a sub-class of location problems (specially when we refer to location problems with distance constraints as in Moon & Chaudhry, 1984), and we can find nowadays many researchers specifically devoted to them. In this sense, we may say that maximizing diversity can be considered now as a field in itself.

As recently pointed out by Parreño, Álvarez-Valdés, & Martí (2021), the term diversity is somehow ambiguous in the context of combinatorial optimization, and it has been applied to problems looking for dispersion among the selected points, but also in problems looking for some kind of representativeness, in which the

* Corresponding author.
E-mail addresses: rafael.marti@uv.es (R. Martí), gavara@uv.es (A. Martínez-Gavara), sergio.perez.pelo@urjc.es (S. Pérez-Peló), jesus.sanchezoro@urjc.es (J. Sánchez-Oro).

https://doi.org/10.1016/j.ejor.2021.07.044

selected points are class representatives of subsets of points in the given set. This argument is not entirely new, since Glover, Kuo, & Dhir (1998) in the late nineties already said that *diversity is a rather nebulous term with overtones and a vaguely statistical nature*, and proposed simple heuristics that could easily be adapted to handle the particular characteristics of the diversity problem being solved.

Maximum diversity problems have a wide variety of real-life applications that cover many fields. One of the first applications appears in genetics (see Porter, Rawal, Rachie, Wien, & Williams, 1975) where species with desirable traits are selected to obtain new varieties by controlled breeding. These problems can also be applied to other areas related to biology, such as ecology (Pearce, 1987) where diversity is crucial to establish viable systems. The selection of a diverse group as a representative sample is probably one of the most extended applications which arises in product design (Glover et al., 1998), ethnicity (Swierenga, 1977), and in making diverse teams at work. The placement of undesirable facilities such as hazardous waste sites, and location problems with associated capacity and cost factors, have been also studied as diversity maximization problems by several researchers (see Church & Garfinkel, 1978; Erkut & Neuman, 1989; Goldman & Dearing, 1975; Rosenkrantz, Tayi, & Ravi, 2000 and the references cited therein).

We have identified three periods in the development of diversity and dispersion problems. The **early period**, from 1977 to 2000, where we can find the first models (MaxMin and MaxSum), and relatively simple algorithms to solve them, being the seminal papers by Kuby (1988) and Erkut (1990) the origins of the area. We can only find a few papers in this period in the OR literature, although in other fields of science, such as sociology or biology, diversity maximization received much more attention.

In the second period, that we may call the **expansion period**, the first metaheuristics were proposed to target large instances effectively. Duarte & Martí (2007) adapted both the Tabu Search and GRASP methodologies to the MaxSum model, triggering the interest of the metaheuristic community in this family of problems. Special mention deserves the work by Prokopyev, Kong, & Martinez-Torres (2009), where three new dispersion models were introduced: the MaxMinSum, the MaxMean, and the MinDiff. In this way, these authors clearly stated that there are different ways to model diversity maximization, opening many possibilities for future developments. This period lasted over a decade, ending with very efficient methods for some of the models, as shown in the empirical comparison of 30 methods by Martí, Gallego, Duarte, & Pardo (2013) performed in 2010, and with several solid research groups working on them. The boundaries defining the area of maximizing diversity were expanded with the inclusion of more realistic models built with capacity and cost constraints.

The third period, that we call the **development period**, started in 2011 and is still in progress. From the heuristic side, the competition is now very high, due to the efficient methods published in the previous period, so only complex metaheuristics are proposed now. In the exact domain, Sayyady & Fathi (2016) and Sayah & Irnich (2017) recently proposed integer programming approaches for the MaxMin model, which are able to solve large size problems, and somehow changed the game in terms of the need of heuristics for real instances. These new efficient methods, exact and metaheuristics, made Martí's comparison (Martí et al., 2013) out of date, so one of the objectives of this paper is to update it by including them.

Martínez-Gavara, Corberán, & Martí (2021) elaborated on the seminal work by Rosenkrantz et al. (2000) that included capacity and cost constraints in the classic diversity models. The authors approach these theoretical models from an Operations Research perspective, opening new research opportunities and modeling a wide range of real problems. In this paper, we complete their proposals by introducing other variants that may be the subject of future developments as well.

Most of the studies on diversity problems have been computational, and the different methods have been tested on a well-established benchmark set of instances. The Maximum Diversity Problem Library, MDPLIB, was originally collected for the MaxSum model, and contained 315 instances proposed and used in the development period. This library has been used to evaluate heuristics for all dispersion models. However, some type of instances are not well suited for some of the models and, additionally, some of them are trivial for nowadays complex methods. We therefore revised it, removing some small instances, and adding some new ones, specifying the models for which they are meant. We call MDPLIB 2.0[1] to the updated library that contains 770 instances.

There is no doubt that maximizing diversity is nowadays a trending area in many fields of science. Terms like biodiversity, heterogeneous workforce, or simply gender diversity have a positive connotation and are studied in many disciplines. We obviously do not cover them directly in this paper, but our aim is to show that advances in mathematical models related to diversity have a huge impact in many other disciplines. Researchers in Operations Research perfectly know the power and wide scope of models, but we want to emphasize it here because diversity is a cross cutting concept, which makes these models applicable to many areas. This point is clearly stated in a management science paper (Hong & Page, 2004) directly entitled as *Groups of diverse problem-solvers can outperform groups of high-ability problem solvers*, thus reinforcing the idea that maximizing diversity has benefits even in problem solving. In the following sections, we review the contributions to discrete diversity optimization classifying them into the three periods introduced above. We basically consider models, solving methods, and benchmark instances. We finish our revision with an empirical comparison of the two most studied models, the MaxSum and the MaxMin, and a recently considered combination of them.

2. The early period (1980–2000)

Early papers on diversity and dispersion problems can be traced back to the late seventies. It seems that Shier (1977) was the first to recognize the *p*-dispersion as an optimization problem. He considered the continuous problem of locating a facility at a node or any point in the arcs of a tree. Chandrasekaran & Daughety (1981) studied the *p*-center and *p*-dispersion[2] discrete problems on a tree. The *p*-center minimizes the maximum distance between the selected nodes in a tree, while the *p*-dispersion maximizes their minimum distance. The *p*-center problem had been studied in the previous decade and it was relatively well-known in location theory; however, as the authors mentioned, the *p*-dispersion had received very little attention in spite of its practical significance to model the location of undesirable facilities. The authors studied the duality between both problems.

As far as we know, **the first publication** on discrete versions of dispersion problems in general graphs is due to Kuby (1988). The author considered the *p*-dispersion as locating *p* facilities on the nodes of a network, so that the minimum distance between any pair of facilities is maximized. Kuby proposed a linear integer formulation for this problem and applied it to a small example with 25 nodes. The author also extended the model to the max-sum case, in which the objective is to maximize the sum of distances between all the pairs of selected facilities (nodes). These problems were later coined as the MaxMin Diversity Problem (MMDP), and MaxSum Diversity Problem (MDP) respectively.

[1] Martí, R., A. Duarte, A. Martínez-Gavara, and J. Sánchez-Oro. MDPLIB 2.0 - Maximum Diversity Problem Library. https://www.uv.es/rmarti/paper/mdp.html.

[2] Note that some authors use *p* and others *m* to denote the number of elements to be selected. In this paper we will use both indistinguishably.

The MDP can be trivially formulated in mathematical terms as a quadratic binary problem, where variable x_i takes the value 1 if element i is selected and 0 otherwise, $i = 1, \ldots, n$.

$$\begin{array}{ll} \text{Maximize} & \sum_{i<j} d_{ij} x_i x_j \\ \text{subject to:} & \sum_{i=1}^{n} x_i = m \\ & x_i \in \{0,1\} \qquad i = 1, \ldots, n. \end{array} \tag{1}$$

To avoid the non-linearity due to the product of two variables, Kuby formulated the MDP as:

$$\begin{array}{lll} \text{Maximize} & \sum_{i<j} z_{ij} d_{ij} & \\ \text{subject to:} & \sum_{i=1}^{n} x_i = m & \\ & z_{ij} \le x_i & i, j = 1, \ldots, n : j > i. \\ & z_{ij} \le x_j & i, j = 1, \ldots, n : j > i. \\ & z_{ij}, x_i \in \{0,1\} & i, j = 1, \ldots, n. \end{array} \tag{2}$$

This author also formulated the m-dispersion problem (MMDP) in the following terms, where C is a very large constant number that makes the second constraint active only when facilities i and j have been selected ($x_i = x_j = 1$):

$$\begin{array}{lll} \text{Maximize} & D & \\ \text{subject to:} & \sum_{i=1}^{n} x_i = m & \\ & D \le d_{ij}(1 + C(1 - x_i) + C(1 - x_j)) & i, j = 1, \ldots, n : j > i. \\ & x_i \in \{0,1\} & i = 1, \ldots, n. \end{array} \tag{3}$$

Erkut and Neuman published in 1989 an invited review in the European Journal of Operational Research on location models for obnoxious facilities (Erkut & Neuman, 1989), where a function distance is maximized. The authors mainly focused on continuous and network based models, and pointed out that the only previous work on discrete models is the one by Kuby described above. The authors classified these models according to the following criteria:

- number of facilities (single / multiple)
- solution space ($\mathbb{R}^k$ / network)
- feasible region (discrete / continuous)
- distance measure (Euclidean / rectilinear / network)
- weights (different weights / unweighted)
- distance function (sum / min)
- objective (single / multiple)

Erkut (1990) proposed the first algorithms for the MaxMin. In particular, this author introduced a simple heuristic and a branch and bound exact method to solve small problems (with up to 40 elements) to optimality. Erkut's heuristic is intentionally naïve and breaks ties in the constructive phase at random. As document by Hart & Shogan (1987), semi-greedy heuristics, which deviate from rigid selection rules by including random choices, generate many solutions, leading to better outcomes than simple greedy heuristics. This is very interesting since this type of reasoning led to the design of powerful metaheuristics, such as the well-known GRASP methodology (Feo & Resende, 1995; Festa & Resende, 2016). The construction is coupled with a local search method that scans the set of selected elements in search of the best exchange to replace a selected element with an unselected one. The method performs moves as long as the objective value increases, and stops when no improving exchange can be found. This local search has been applied to most of the algorithms proposed for both the MaxSum (MDP) and the MaxMin (MMDP), introducing successive refinements that resulted in improved outcomes.

Kincaid (1992) proposed two heuristics for the MaxMin, also known as the discrete p-dispersion problem, based on exchanges: a simulated annealing (SA) heuristic (Kirkpatrick, Gelatt, & Vecchi, 1983) and a tabu search (TS) heuristic (Glover, Campos, & Martí, 2021; Glover & Laguna, 1998). In a given iteration, these heuristics generate a random move (an exchange between a selected and an unselected element) and apply the standard acceptance rules of the methodology, the so-called temperature and cooling schedule in the SA, and the tabu status and aspiration criteria in tabu search. These methods are also adapted to the MaxSum problem, called the p-defense-sum problem in that paper. The author examined the performance of both methods on these two models on a reduced benchmark of 30 instances of size $n = 25$ (in three groups of ten with different characteristics) and p ranging from 5 to 15.

Kuo et al. (1993) proposed several models to maximize diversity based on their seminal working papers elaborated in 1977. Independently to Kuby and Erkut, the authors presented some efficient binary programming models for the MaxSum and MaxMin. In particular, for the MaxSum, called there the Maximum Diversity Problem, they proposed the following zero-one formulation that has been considered the most efficient one until now:

$$\begin{array}{lll} \text{Maximize} & \sum_{i<n} w_i & \\ \text{subject to:} & \sum_{i=1}^{n} x_i = m & \\ & -U_i x_i + w_i \le 0 & i = 1, \ldots, n-1. \\ & -\sum_{j=i+1}^{n} d_{ij} x_j + L_i(1 - x_i) + w_i \le 0 & i = 1, \ldots, n-1. \\ & x_i \in \{0,1\} & i = 1, \ldots, n. \end{array} \tag{4}$$

where $U_i = \sum_{j=i+1}^{n} \max(0, d_{ij})$ and $L_i = \sum_{j=i+1}^{n} \min(0, d_{ij})$. The authors proved that the MDP is NP-hard both with and without restricting distances to non-negative values. Kuo, Glover, and Dhir also proposed a binary model for the MaxMin, and illustrated its performance on a small example of size 10. The same authors applied these models to solve a practical case in biological diversity (Glover, Kuo, & Dhir, 1995).

Ghosh (1996) proved that the MaxMin problem is NP-hard using a reduction from the vertex cover problem. The author proposed a greedy randomized heuristic, that can be considered the first step of extending simple heuristics to complex metaheuristic, and presented a limited computational experience on small instances (up to $n = 40$) to show its merit.

Glover et al. (1998) proposed four different heuristics for the MaxSum problem. The authors highlighted that different versions of this problem include additional constraints, so the objective is to design heuristics whose basic moves for transitioning from one solution to another are both simple and flexible, allowing these moves to be adapted to multiple settings. In this line, they consider moves that are especially attractive in this context: constructive and destructive, that drive the search to approach and cross feasibility boundaries. These type of moves are natural in the maximum diversity problem, where the goal is to determine an optimal composition for a set of selected elements. The authors compare the solutions obtained with their heuristics with the optimal solutions in small instances (up to $n = 30$), and conclude that the constructive method C2, and the destructive method D2 perform very well considering their simplicity. Specifically, C2 starts by randomly selecting an initial element. Then, it selects at each step, the element with the maximum sum of distances to the already selected elements. On the other hand, D2 starts with all the elements selected, and deselects the element with the minimum distance to the selected elements at each step. Both methods finish when m elements are selected.

At the end of this period, Ağca, Eksioglu, & Ghosh (2000) proposed a Lagrangian approach and provided both lower and upper bounds for the MaxSum problem. The authors also proposed a variation of their method to target the MaxMin problem. Extensive experimentation with small size instances (up to $n = 100$) showed the good quality of the results in comparison with previous heuris-

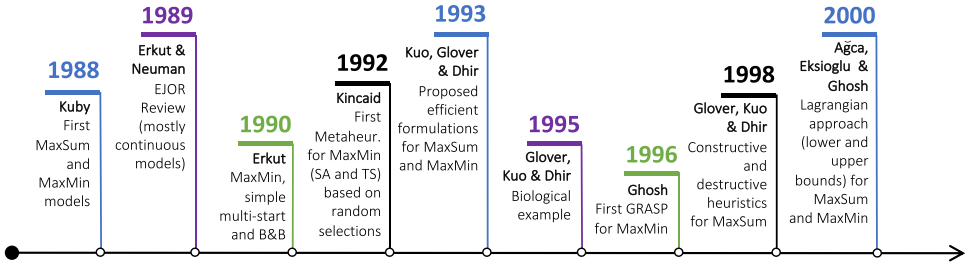

Fig. 1. Timeline of early OR diversity contributions (1988–2000).

tics; however, as the authors admit, this comes with a cost of significant longer running times.

Fig. 1 shows a timeline diagram, in which the main contributions in this early period are depicted.

3. The expansion period (2000–2010)

This second period witnessed a huge growth in the area. Most of the efforts were devoted to the MaxSum model, with a total of 30 methods, as documented in Martí et al. (2013). The MaxMin model on the other hand, also received attention, although moderate, probably because it poses a challenge to heuristic methods due to the flat landscape in the search space created by the combination of the maximum and minimum in the objective function. Finally, new models were also proposed, introducing new ways to compute diversity and including constraints to target more realistic variants. We call it the expansion period since the limits defining the area were substantially expanded in this decade.

From a graph-theoretic point of view, we may highlight the work by Chandra & Halldórsson (2001) in which many dispersion problems are classified in terms of its nodes and edges. In particular, given a graph G with set of vertices V, and an integer value p, a dispersion problem can be defined as obtaining a set of p vertices $P \subseteq V$ in a way that the sum of the weights of certain edges in the subgraph induced by P is maximized. The edge set definition characterizes the dispersion problem. It includes cliques, trees, k-trees, stars, pseudo-forests, cycles or matchings. The authors proved the NP-hardness of these problems and proposed simple heuristics with performance approximation ratios depending of the metric of the space. In line with that, Fekete & Meijer (2004) consider the MaxSum problem, called the heaviest subgraph problem, in the case of d-dimensional spaces with rectilinear distances, and establish a linear-time algorithm that finds an optimal solution. In this way, they improve upon the best known result so far of an approximation algorithm with performance ratio 2.

From the practical side, most of the papers published consider the complete subgraph induced by the selected points. This is specially true in the metaheuristic field, as shown below.

3.1. The MaxSum model

Many metaheuristic methodologies were implemented in this period to the MaxSum problem. They were relatively simple at the beginning, but as the competition among methods became harder, more sophisticated search strategies were proposed, ending up with very complex algorithms. GRASP, Tabu Search, and VNS played an important role in this period.

As mentioned, Kincaid (1992) was the first to apply metaheuristics, namely SA and K−TS, to the MaxSum problem, although they were straightforward implementations. Macambira (2002) proposed a similar implementation of the tabu search methodology, called M−TS, to solve the MDP. Note that K−TS starts with a random solution while M−TS starts with a greedy constructed solution.

Silva, Ochi, & Martins (2004) proposed several heuristics based on the **GRASP** methodology (Feo & Resende, 1995). They combined different constructions with local searches and tested them on a wide set of instances, which includes the largest reported so far. They called KLD to the basic construction algorithm, and KLDv2 and MDI to the improved versions of KLD. It must be noted that in these largest instances with $n = 500$ elements, the methods require many hours of running time. Santos, Ribeiro, Plastino, & Martins (2005) presented a hybrid method, GRASP−DM, combining GRASP with data mining techniques, which basically consists of two phases. First, the GRASP phase is executed a certain number of iterations. Then, the data-mining process extracts patterns from an elite set of solutions that guide the following GRASP iterations. Silva, De Andrade, Ochi, Martins, & Plastino (2007) revisited the problem to propose a hybrid method, GRASP−PR, combining GRASP with Path Relinking (Laguna & Martí, 1999). As in the hybrid method above, an elite set is populated with the solutions obtained with the application of a GRASP algorithm. Then, path relinking is applied from each solution in the elite set (initial solution) to the local optimum obtained in each new GRASP iteration (guiding solution). In this way, the method creates a path by adding elements in the guiding solutions to the initial solution (and dropping those not present in the guiding solution).

As far as we know, the work of Katayama & Narihisa (2006) is the only one where a standard **Memetic Algorithm** (MA) is applied in this period. The algorithm combines a randomized greedy construction method with an evolutionary algorithm, a repair mechanism to guarantee the feasibility of the solutions, and a local search. Aringhieri & Cordone (2006) presented a **Scatter Search** procedure, A−SS, which can be considered a special case of a memetic algorithm. In particular, this method iterates over a small set of elite solutions, instead of the traditional population of a relatively large size, called the reference set, RefSet. In this particular implementation of scatter search, the RefSet is divided into two subsets, one with the best solutions found during the search, and the other one with solutions that largely differ from each other and from the best ones. Gallego, Duarte, Laguna, & Martí (2009) proposed an alternative scatter search algorithm, G−SS, for the MaxSum problem. In their approach, the distance between solutions is used to measure how diverse one solution is with respect to a set of solutions. The method applies a tabu search algorithm to improve the combined solutions, thus creating a hybrid of a memetic algorithm with a tabu search. It is very interesting that in the following decade, that we call the development period, most of the proposed algorithms for the MaxSum follow this scheme of combining these two methodologies, which will emerge as the best choice for this problem.

In the domain of exact methods, the first important approach was due to Pisinger (2006), who proposed upper bounds for both MaxSum and MaxMin problems. Based on that bounds, **Branch and Bound** methods were respectively derived. The experiments

showed that in the MaxSum problem, the method is able to solve the medium size Euclidean instances with $n = 80$ in about 500 seconds, but it encounter difficulties to find the optimum in the random instances, in which requires more than 3 hours in those with $n = 60$. A similar situation is described for the MaxMin solver, in which both types of instances are solved when $n < 80$. This branch and bound clearly outperforms the first one proposed by Erkut (1990).

Duarte & Martí (2007) applied two metaheuristics for the MaxSum problem. Specifically, the authors introduced a **Tabu Search**, LS-TS, and two GRASP algorithms, called GRASP-C2 and GRASP-D2, and proposed several strategies to explore the typical neighborhood based on exchanges in an efficient way, to avoid the long running times of previous tabu search and GRASP implementations. In particular, instead of searching for the best exchange at each iteration, their neighborhood exploration performs two stages. In the first one, it selects the element with the lowest contribution to the value of the current solution. Then, in the second stage, the method performs the first improving move to replace it (i.e., instead of scanning the whole set of unselected elements searching for the best exchange, it performs the first improving exchange without examining the remaining unselected elements). Their experimentation confirms the effectiveness of the proposed strategies.

Palubeckis (2007) proposed an Iterated Tabu Search, ITS, that alternates tabu search with perturbation procedures. Aringhieri, Cordone, & Melzani (2008) presented XTS, a tabu search with short and long term memory functions such as LS-TS. A novelty of this method is that the tabu tenure parameter is dynamically set during the execution of the algorithm (i.e., it is increased if the solution value has steadily improved, and it is reduced if the solution value has steadily worsened). Aringhieri & Cordone (2011) proposed a random re-start method, RR, which constructs an initial solution with a greedy procedure similar to the simple method proposed by Erkut (1990). Then, the constructed solution is improved by means of a simplified version of XTS.

Variable Neighborhood Search (Hansen & Mladenović, 2005) (VNS) was applied to the MaxSum problem too. As it is well-known, this methodology is based on a simple and effective idea, a systematic change of the neighborhood within a local search algorithm, and proved to be the best option to solve the MaxSum problem at that time.

Silva et al. (2004) proposed a simple VNS, SOMA, based on two neighborhoods. It first applies the classic local search (Ghosh, 1996) until no further improvement is possible. Then, a second local search based on swapping two elements in the solution by another two not present in the solution is performed. Brimberg, Mladenović, Urošević, & Ngai (2009) proposed several VNS procedures originally devoted to the heaviest k-subgraph problem, which generalizes the MDP. The authors presented a skewed VNS, a basic VNS (B-VNS), and a combination of a constructive heuristic followed by VNS. The best variant is B-VNS and consists of three main elements. The first one, called Data Structure, allows the algorithm to efficiently update the value of the objective function; the second one, Shaking, generates solutions in the neighborhood of the current solution by performing random vertex swaps; and the third one is a local search procedure based on exchanges.

Aringhieri & Cordone (2011) presented four VNS implementations: Basic VNS, Guided VNS, Accelerated VNS, and Random VNS. An important characteristic is their hybridization with Tabu Search to locally improve the generated solutions. Accelerated VNS, A-VNS, seems to be the best variant, and it makes re-starts much less frequent because the number of neighborhoods is considerably larger than the values used in the Basic and Guided variants.

Martí, Gallego, & Duarte (2010) proposed a branch and bound algorithm for the MaxSum problem. The authors considered an implicit enumeration of the solutions (selections of m elements), and compute upper bounds for partial solutions. Their method is embedded in the standard search tree to fathom the nodes (subsets of solutions defined by a partial selection), thus discarding for examination many nodes in the search tree. This combinatorial branch and bound solves small instances easily ($n = 50$), most of the medium instances with $n = 100$, and cannot solve the large ones considered ($n = 150$) in 1 hour of CPU time.

We close this period on the MaxSum problem with an empirical comparison of all the methods published so far, performed in 2010 (although published a few years later). Martí et al. (2013) presented an extensive computational experimentation to compare 10 heuristics and 20 metaheuristics for the MaxSum problem, most of them summarized in Table 1.

Martí et al. (2013) proposed the first version of the so-called MDPLIB in which they collected 315 instances introduced by different authors in previous papers. Their empirical comparison with 30 methods was exhaustive, and concluded with the final comparison of the five methods identified as the best ones over two time horizons, 10 and 600 seconds of CPU time. We reproduce here their final table in Table 2 with the results, average percent deviation (% *dev*) and number of best solutions (# *best*), of the best GRASP method, GRASP-D2, the best local search based methods, which includes a tabu search, ITS, and two variable neighborhood search, A-VNS and B-VNS, and the best population based method, G-SS.

As expected, the average percentage deviations of the methods are lower when the CPU time increases from 10 seconds to 600 seconds. In this way, after 600 seconds of CPU time, the five methods under comparison present deviations lower than 1%. In line with this, the number of best solutions found increases as the running time increases. The Friedman test confirms the superiority of the VNS based methods, from which B-VNS emerges as the best method overall, followed by the tabu search ITS as the second best.

3.2. The MaxMin model

As described in the previous section, after Kuby's seminal paper (Kuby, 1988), Erkut (1990) proposed a simple heuristic, Kincaid (1992) a simulated annealing and a tabu search, and Ghosh (1996) a multi-start heuristic. Although Kincaid's heuristics are based on complex methodologies, his algorithms are straightforward implementations, in which the neighborhood is scanned by random sampling. On the other hand, the multi-start by Ghosh examines the entire neighborhood in the local search, implementing the so-called best strategy. In contrast, Resende, Martí, Gallego, & Duarte (2010) applied the GRASP methodology to the MaxMin problem, but with an efficient implementation that is able to obtain high-quality solutions in short running times, outperforming all previous developments. We describe now this method in detail since it was the best for the MaxMin in this period.

Given a set N with n elements, the construction procedure in Resende et al. (2010) performs m steps to produce a solution with m elements. The set *Sel* represents the partial solution under construction. At each step, the constructive method selects a candidate element $i^* \in CL = N \setminus Sel$ with a large distance to the elements in the partial solution *Sel*. Specifically, it first computes d_j as the minimum distance between element j and the selected elements. Then, it constructs the restricted candidate list *RCL* with all the candidate (unselected) elements j with a distance value d_j within a fraction α ($0 \le \alpha \le 1$) of the maximum distance $d^* = \max\{d_j \mid j \in CL\}$. Finally, the method randomly selects an element in *RCL*.

Table 1
Metaheuristics for MaxSum in 2010.

Methodology	Algorithms	References
Simulated Annealing	SA	Kincaid (1992)
GRASP	KLD, KLDv2, MDI, GRASP-DM, GRASP-C2, GRASP-D2, GRASP-PR	Silva et al. (2004), Santos et al. (2005), Duarte & Martí (2007), Silva et al. (2007)
Tabu Search	K-TS, M-TS, LS-TS, ITS, XTS, RR	Kincaid (1992), Macambira (2002), Duarte & Martí (2007), Palubeckis (2007), Aringhieri et al. (2008), Aringhieri & Cordone (2011)
VNS	SOMA, B-VNS, A-VNS	Silva et al. (2004), Brimberg et al. (2009), Aringhieri & Cordone (2011)
Scatter Search	A-SS, G-SS	Aringhieri & Cordone (2006), Gallego et al. (2009)
Memetic Algorithms	MA	Katayama & Narihisa (2006)

Table 2
Best MaxSum methods on MDPLIB instances in 2010.

CPU		GRASP-D2	A-VNS	B-VNS	ITS	G-SS
10 seconds	% *dev*	1.57	0.16	0.08	0.17	0.24
	# *best*	10	60	51	51	51
600 seconds	% *dev*	0.63	0.03	0.02	0.02	0.13
	# *best*	32	75	83	62	59

Given a set N with n elements, and a solution Sel with m selected elements, we can compute the following values:

$$d_i = \min_{j \in Sel} d_{ij}, \quad d^* = \min_{i \in Sel} d_i,$$

where d_i is the minimum distance of element i to the selected elements (those in Sel), and d^* is the objective function of the current solution. It is clear that to improve a solution we need to remove (and thus replace) the elements i in the solution for which $d_i = d^*$.

The local search method in Resende et al. (2010) scans, at each iteration, the list of elements in the solution ($i \in Sel$) with minimum d_i value, i.e. for which $d_i = d^*$, starting with a randomly selected element. Then, for each element i with a minimum d_i-value, the local search examines the list of unselected elements ($j \in N \setminus Sel$) in search for the first improving exchange. The unselected elements are also examined in lexicographical order, starting with a randomly selected element. The method performs the first improving move ($Sel \leftarrow Sel \setminus \{i\} \cup \{j\}$) and updates d_i for all elements $i \in Sel$ as well as the objective function value d^*, concluding the current iteration. The algorithm repeats iterations as long as improving moves can be performed and stops when no further improvement is possible.

An important characteristic of this GRASP for the MMDP is the definition of improving move. To efficiently search the flat landscape of the MaxMin problem, the authors introduced in the local search an extended meaning of the term improving. In particular, a move is considered to improve the current solution if it increases the value of d^*, or keeps d^* fixed and reduces the number of elements i with $d_i = d^*$. The method stops when no further improvement is possible according to this definition.

The GRASP method above is coupled with a Path Relinking (PR) post-processing for improved outcomes. The PR algorithm operates on a set of solutions, called *elite set* (*ES*), constructed with the best solutions obtained with GRASP. It basically creates paths of solutions between elite solutions. Let x and y be two solutions, PR starts with the first solution x, and gradually transforms it into the second one y, by swapping out elements selected in x with elements selected in y. The elements selected in both solutions x and y remain selected in the intermediate solutions generated in the path between them. The output of each PR iteration is the best solution, different from x and y, found in the path.

Resende et al. (2010) compiled a benchmark library of instances reported in the previous papers on the MaxMin problem to perform an empirical comparison among the heuristics. In particular, they considered three sets of instances named *Glover*, *Geo*, and *Ran*. The first one includes small Euclidean instances ($n \leq 30$) from randomly generated points in a multi-dimensional space. The second one, *Geo*, extends the first one by including larger instances ($100 \leq n \leq 500$). The third one, *Ran*, consists of large matrices with integer random numbers. These sets are include in the MDP Library of Benchmark Instances described in Section 5.

Table 3
Best MaxMin methods on *Geo* instances in 2010 .

		Multi-Start	SA	TS	GRASP	GPR
$n = 100$	% *dev*	0.75	0.00	0.00	0.76	0.09
	# *best*	10	19	20	10	17
	time (*s*)	2.45	20.96	33.64	0.68	3.76
$n = 250$	% *dev*	1.00	0.68	1.75	1.11	0.16
	# *best*	0	6	2	1	14
	time (*s*)	30.50	220.57	439.68	5.58	65.57
$n = 500$	% *dev*	2.36	3.48	9.27	2.39	0.04
	# *best*	0	0	0	0	16
	time (*s*)	282.37	1449.85	3633.36	34.99	1465.44

0.00 means less than 0.001

We do not reproduce here the entire analysis in Resende et al. (2010), but we show in Table 3 the comparison of their GRASP, and GRASP with Path Relinking (GPR), with the Multi-Start method by Ghosh (1996), Simulated Annealing (SA) and Tabu Search (TS) by Kincaid (1992). This table shows, for each method, the average relative percentage deviation (% *dev*) between the best solution value obtained with that method and the best known value for that instance. It also reports, for each method, the number of instances (# *best*) in which the value of the best solution obtained with this method matches the best known value. Finally, it reports the associated running times in seconds on a Pentium 4 computer running at 3 GHz.

Results from Table 3 has to be interpreted with caution because we are comparing, at the same time, methodologies and implementations. This is probably the weakness in the computational comparison of heuristic papers. It is very difficult to evaluate how much of the solution's quality is due to the methodology, and how much to the specific way in which it is implemented to solve a problem. Note that implementation not only includes search strategies in the solution space, but also data structures management, and even computer language. For example, GRASP obtains better results than TS in the large instances in this table ($n = 500$), with 2.39% and 9.27% average deviations respectively. However, in the small instances ($n = 100$) we observe the opposite situation, since GRASP has an average deviation value of 0.76% and TS has a value lower than 0.001%. This seems to indicate that the implementation strategies, that usually play an important role in large instances, may be responsible for this difference. In our opinion, we cannot conclude from this type of experiment that one methodology is better than the other one, and we can only state that this GRASP implementation performs better in large instances than this Tabu Search implementation.

3.3. Other diversity models

Rosenkrantz et al. (2000) introduced several diversity models constrained in terms of cost and capacity, motivated by their practical applications in facility location. For example, the location of undesirable or hazardous facilities, such as waste sites or nuclear plants, requires their dispersion while satisfying a certain total demand. Another example can be found in the context of retail franchises, where stores should not be located close to each other. Facilities and stores have a capacity to provide a service in systems that require an overall demand, and it is clear in practical terms that they have an associated setup or operational cost, which makes appropriate to consider a certain limit in the total expenses generated. As stated by the authors, "these practical aspects add a new dimension to the conventional dispersion problem". Classical models, such as the MaxSum or MaxMin, indirectly address the problem requirements by considering a pre-fixed number of facilities (i.e., the number of points to be selected is an input to the problem). However, this simplification is not realistic in many settings.

The work by Rosenkrantz et al. (2000) was mainly theoretical. The authors proposed different models to tackle diversity, capacity and cost, where one of them is optimized (plays the role of the objective function), and the other two are included as constraints. Specifically, the three variants proposed were:

(i) maximize capacity under distance and cost constraints (Max-Cap/Dist/Cost),
(ii) minimize cost under capacity and distance constraints (Min-Cost/Cap/Dist),
(iii) maximize distance under capacity and cost constraints (Max-Dist/Cap/Cost).

When the capacity is a constraint, the authors introduced a minimum capacity B reflecting the required level of service. Similarly, when the cost is a constraint, a maximum budget K is considered. The authors also introduced two models with distance and capacity (Max-Cap/Dist and Max-Dist/Cap). Rosenkrantz et al. (2000) established the NP-hard complexity of these variants, proved the existence of an approximate algorithm within a factor 2 in the Max-Dist/Cap with distances satisfying the triangle inequality, and the non-approximability results for the other variants. In particular, they provided proof of the non-existence of a polynomial-time approximation scheme for the Max-Dist/Cap/Cost variant, and proposed a greedy heuristic based on binary search for the Max-Dist/Cap problem. Although no empirical results or experiments are reported, the theoretical study concludes that their heuristic running time is $O(n^2 \log(n))$.

Surprisingly, in spite of its potential impact, this paper was ignored by the metaheuristic community at that time, and we had to wait until the next decade to see the first complex heuristics for these new problems.

Prokopyev et al. (2009) introduced four additional dispersion models, combining and generalizing the well-known MaxSum and MaxMin models. The *MaxMean Dispersion Problem* (Max-Mean) that maximizes the average instead of the sum, can be formulated as the following 0–1 integer linear programming problem:

$$\begin{aligned} \text{Maximize} \quad & \frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij}x_i x_j}{\sum_{i=1}^{n} x_i} \\ \text{subject to} \quad & \sum_{i=1}^{n} x_i \geq 2 \\ & x_i \in \{0,1\} \qquad i = 1,\ldots,n. \end{aligned} \tag{5}$$

An interesting characteristic in the MaxMean model (5), is that the cardinality restriction is not imposed, and a solution may be formed by an arbitrary number of elements. In this sense we can say that this model generalizes the MaxSum model, since the number of elements to be selected is not set beforehand, and the model selects it when maximizing the objective. A further generalized version of this problem introduces weights associated to the nodes. It is called *Generalized MaxMean Dispersion Problem* and is formulated as follows:

$$\begin{aligned} \text{Maximize} \quad & \frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij}x_i x_j}{\sum_{i=1}^{n} w_i x_i} \\ \text{subject to} \quad & \sum_{i=1}^{n} x_i \geq 2 \\ & x_i \in \{0,1\} \qquad i = 1,\ldots,n. \end{aligned} \tag{6}$$

where w_i is the weight assigned to element $i \in V$.

Prokopyev et al. (2009) introduced two other models in the context of diversity called *equity models*, which incorporate the concept of fairness among candidates. These models appear in different settings, such as urban public facility location, diverse/similar group selection, and sub-graph identification, in which one may address fair diversification or assimilation among members of a network. The MaxMinSum diversity problem maximizes the minimum aggregate dispersion among the chosen elements, while the Minimum Differential Dispersion model, MinDiff, minimizes extreme equity values of the selected elements.

The *Maximum MinSum Dispersion Problem*, MaxMinSum, consists of selecting a set $M \subseteq V$ of m elements such that the smallest total dispersion associated with each selected element i is maximized. The problem is formulated in Prokopyev et al. (2009) as follows:

$$\begin{aligned} \text{Maximize} \quad & \left\{ \min_{i:x_i=1} \sum_{j:j\neq i} d_{ij}x_j \right\} \\ \text{subject to} \quad & \sum_{i=1}^{n} x_i = m \\ & x_i \in \{0,1\} \qquad i = 1,\ldots,n. \end{aligned} \tag{7}$$

The Minimum Differential Dispersion model, MinDiff, is probably the most elaborated one in terms of its objective function definition. It basically consists of computing the maximum and minimum total dispersion associated to the m selected elements, minimizing their difference. In this way, we obtain a balance selection of elements in the sense that their associated dispersion values are very similar, and this is why it is introduced as an equity model. This problem can be formulated in simple terms as follows, although more efficient formulations are proposed in Prokopyev et al. (2009).

$$\begin{aligned} \text{Minimize} \quad & \left\{ \max_{i:x_i=1} \sum_{j:j\neq i} d_{ij}x_j - \min_{i:x_i=1} \sum_{j:j\neq i} d_{ij}x_j \right\} \\ \text{subject to} \quad & \sum_{i=1}^{n} x_i = m \\ & x_i \in \{0,1\} \qquad i = 1,\ldots,n. \end{aligned} \tag{8}$$

Table 4 collects the diversity models introduced so far. For the sake of simplicity we do not include the weighted MaxMean proposed by Prokopyev et al. (2009), and the variations of capacity and cost in Rosenkrantz et al. (2000).

It is worth mentioning the connection between diversity models and Unconstrained Binary Quadratic Programming (UBQP). As described in the survey by Kochenberger et al. (2014), UBQP refers to a relatively simple model that represents a wide range of problems, from facility location to partitioning. Diversity problems fit

Table 4
Diversity models.

Problem		Obj. function	Constraints	Cardinality	Context
MaxSum	Kuby (1988)	$\sum_{i<j,i,j\in M} d_{ij}$	$\|M\| = m$	fixed	Diversity
MaxMin	Kuby (1988)	$\min_{i<j,i,j\in M} d_{ij}$	$\|M\| = m$	fixed	Dispersion & Equity
MaxMin/Cap/Cost	Rosenkrantz et al. (2000)	$\min_{i<j,i,j\in M} d_{ij}$	$CAP(M) \geq B\ COST(M) \leq K$	variable	Dispersion & Equity
MaxMean	Prokopyev et al. (2009)	$\frac{\sum_{i<j,i,j\in M} d_{ij}}{\|M\|}$	$\|M\| \geq 2$	variable	Diversity
MaxMinSum	Prokopyev et al. (2009)	$\min_{i\in M} \sum_{j\in M, j\neq i} d_{ij}$	$\|M\| = m$	fixed	Diversity
MinDiff	Prokopyev et al. (2009)	$\max_{i\in M} \sum_{j\in M, j\neq i} d_{ij} - \min_{i\in M} \sum_{j\in M, j\neq i} d_{ij}$	$\|M\| = m$	fixed	Equity

well in that general model, in which the objective function consists of a quadratic expression with the standard form $x'Qx$, where Q is a square matrix (with the distances in the case of diversity problems). Starting with the early mathematical models by Prof. Glover for the MaxSum problem, diversity models and their solution methods have certainly benefited from the extensive research in UBQP. Other researchers in the field, such as Profs. Palubeckis, Hao, or Lõ, worked in both UBQP and diversity models, thus taking advantage of the connections between both models. We believe that these connections will inspire in the near future new ideas for better solving problems of both sides.

4. The development period (2010–2021)

Considering that in the previous decade many methods were proposed for both MaxSum and MaxMin, it is expected that the scientific production in these problems is now moderate in terms of the number of papers but contains very complex methods to compete with the vast existing literature. On the other hand, the other diversity models proposed received very little attention and we will see that researchers are developing now efficient methods for them. It is especially true in the case of restricted models, which, despite being proposed at the beginning of the previous period, had to wait until this one to trigger the interest of researchers.

4.1. The MaxSum model

At the end of the expansion period, Martí et al. (2013) reviewed 30 methods for the MaxSum, compared them on the MDPLIB, and concluded that a tabu search, ITS, and a variable neighborhood search, B-VNS, were the best overall. We have identified in the current period five papers proposing advanced methods that try to improve these two previous methods.

An open question in the heuristic community is if it is better to perform independent constructions, as GRASP typically does, or improved outcomes can be obtained if we use information about past constructions when performing new ones. Lozano, Molina, & García-Martínez (2011) proposed an iterated greedy, IG, for the MaxSum problem, based on this multi-start framework. This method alternates constructive and destructive phases linked by an improvement process. Specifically, after an initial construction, a destruction mechanism removes selected elements, and then reconstructs the partial solution with a greedy method. The resulting solution is improved with a typical local search. An empirical comparison shows that this method is able to obtain solutions of similar quality than the ITS by Palubeckis (2007).

Wang, Zhou, Cai, & Yin (2012) proposed an interesting combination of a Tabu Search with an Estimation of Distribution Algorithm (EDA). The rationale behind this hybrid method, called LTS-EDA, is that the EDA is a knowledge model that implements the information repository in which the experience of the history is stored, to extract the required information by the learnable tabu search for an efficient search exploration of the solution space. Their empirical comparison with previous methods shows that this hybrid method is able to improve previous approaches, especially on large instances. It must be noted that the authors considered very long running times, of 5 hours of CPU time, for the largest instances with $n = 5000$ elements.

Wang, Hao, Glover, & Lü (2014) integrate Tabu Search and Scatter Search in a memetic algorithm. The design of this algorithm is clearly in line with our comments above, that methods in this period are very complex in order to obtain high-quality solutions. In particular, their tabu memetic algorithm, called TS-MA populates an initial reference set with local optima obtained with the application of tabu search to random initial solutions. This tabu search is based on the same neighborhood of previous tabu search implementations for the MaxSum problem, consisting on swapping a selected with an unselected element. However, to reduce the computational effort associated with exploring the neighborhood, they apply a successive filter candidate list strategy, and subdivide the move into its two natural components: first remove an element, and then add another element. The authors explain that one of the key elements in their memetic algorithm is the combination operator based on solution properties by reference to the analysis of strongly determined and consistent variables. The method performs iterations combining the solutions in the reference set as long as the resulting solutions qualify to enter to this set. This method is an improved version of the hybrid metaheuristic published in Wu & Hao (2013). The authors perform an empirical analysis to compare TS-MA with IG (Lozano et al., 2011), ITS (Palubeckis, 2007), B-VNS (Brimberg et al., 2009), and LTS-EDA (Wang et al., 2012). The comparison shows the superiority of the proposed TS-MA; however, it is performed on a limited set of instances, ignoring many instances in the MDPLIB.

De Freitas, Guimarães, Pedrosa Silva, & Souza (2014) proposed a Memetic Self-adaptive Evolution Strategy, MSES. It is basically a population based algorithm that iterates over generations in which parents are mutated to produce children. A strength variable associated with each individual manages the mutation, and it is self-adjusted favoring that best configurations survive over time. As it is customary in memetic algorithms, the method includes a local search and a crossover, and as in previous implementations of the classic exchange-based local search, the authors propose an efficient implementation based on splitting the move evaluation between the removed and the added contribution of its elements. The method is coupled with a tabu search that is selectively applied to the best children in the generation. The algorithm is implemented in Matlab, and it is compared with previous heuristics reimplemented in Matlab as well. The comparison on the MDPLIB instances favors the proposed method.

The last paper published so far on the MaxSum model at the time of writing this review is due to Zhou, Hao, & Duval (2017),

and it describes a memetic algorithm, called OBMA, improved with three search strategies:

- An opposition-based learning to reinforce population initialization as well as the evolutionary search process.
- A tabu search to intensify the search in promising regions.
- A rank-based quality-and-distance pool updating maintain a good level of diversity in the population.

The opposition-based learning basically considers a candidate solution and its corresponding opposite solution. In the case of the MaxSum problem, the opposite solution is simply obtained by selecting some of the elements not selected in a given solution. The tabu search, on the other hand, is based on a constrained swap strategy that manages the size of the explored neighborhood to speed up the method. As all the local search based methods for this problem, it is built upon a swap move that exchanges a selected with an unselected element in the solution. Finally, a rank-pool updating strategy decides whether an improved solution qualifies or not to enter into the population pool in which the memetic algorithm iterates. In particular, this strategy computes a score based on both quality and diversity to rank solutions in the updating process of the pool. The authors compare their OBMA method with five previous methods described above: ITS (Palubeckis, 2007), G-SS (Gallego et al., 2009), B-VNS (Brimberg et al., 2009), IG (Lozano et al., 2011), and LTS-EDA (Wang et al., 2012). The comparison clearly shows that the proposed method consistently obtains the best results in the instances considered. The authors argue that MSES (De Freitas et al., 2014) is not included in this comparison because it is very similar to the TS-MA method. On the other hand, as other empirical comparisons performed in this last decade, it does not consider the entire benchmark of instances published. In Section 6, we perform an exhaustive comparison of the methods identified as the best on the entire MDPLIB benchmark instances.

4.2. The MaxMin model

In this period we have only found two exact methods and one heuristic algorithm for the MaxMin model. These procedures introduce important changes in the way the problem is approached, and therefore they deserve to be described in detail.

Sayyady & Fathi (2016) solve an alternative model consecutively to obtain the optimal solution of the MaxMin model. In particular, they consider the node packing problem, in which given a threshold value l, a graph $G(l)$ is defined with the set V of n nodes of graph $G = (V, E)$, and the set of edges $E(l) = \{(i, j) \in E : d_{ij} < l\}$. The node packing problem consists in finding a maximum cardinality subset of nodes so that no two nodes in this subset are adjacent to each other. It can be formulated in mathematical terms with binary variables, x_i, indicating if node i is selected as:

$$\begin{array}{lll} \text{Maximize} & \sum_{i=1}^{n} x_i & \\ \text{subject to:} & x_i + x_j \leq 1 & i < j, d_{ij} < l \\ & x_i \in \{0, 1\} & i = 1, \ldots, n. \end{array} \tag{9}$$

The authors solve the node packing model above for different values of l. In this way, an optimal solution of the node packing problem in G provides a set of points with minimum distance larger than or equal to l. Note however than in the MaxMin problem, we specifically seek for a set of m points, and the set obtained with the node packing has an arbitrary number of points, called $v(l)$. Sayyady and Fathi proposed to solve a sequence of node packing problems for different values of l according to a binary search, until they obtain a set of $v(l) = m$ points, which turns out to be the optimal solution of the MaxMin model. This method is able to solve large problems to optimality. Specifically, they solve the Euclidean instances with $n = 250$ in less than 200 seconds, and the random instances with $n = 100$ in less than 50 seconds (although they cannot solve the random instances with $n = 250$).

Sayah & Irnich (2017) propose a compact formulation that is able to solve large problems. Let $D^0 < D^1 < \ldots < D^{k_{\max}}$ be the different non-zero distance sorted values in (d_{ij}), and let $E(D^k) = \{(i, j) \in E : d_{ij} < D^k\}$. The location binary variable x_i indicates whether location i is opened, and binary variable z_k indicate whether the location decisions satisfy a minimum distance of at least D^k. Their first formulation follows:

$$\begin{array}{lll} \text{Maximize} & D^0 + \sum_{k=1}^{k_{\max}} (D^k - D^{k-1}) z_k & \\ \text{subject to} & \sum_{i=1}^{n} x_i = m & \\ & z_k \leq z_{k-1} & k = 1, \ldots, k_{\max} \\ & x_i + x_j + z_k \leq 2 & (i, j) \in E(D^k) \setminus E(D^{k-1}) \\ & x_i, z_k \in \{0, 1\} & i = 1, \ldots, n, \ k = 1, \ldots, k_{\max} \end{array} \tag{10}$$

Sayah & Irnich (2017) propose bounds and valid inequalities to strength formulation (10). Their empirical analysis ignores the instances used in previous diversity paper, and considers the *pmed* instances in the OR library. Results are, on the other hand, impressive, since they are able to solve to optimality instances with up to $n = 900$ elements.

Porumbel, Hao, & Glover (2011) proposed a fast local search for a model that combines the MaxMin and the MaxSum problems. In particular, the authors minimize the MaxMin objective function and consider the MaxSum as a secondary objective. The inclusion of this secondary objective is motivated by the fact that there may be a relative large number of solutions that qualify as optimal for the MaxMin, and it makes sense to choose the best one among them in terms of the MaxSum objective. Although not mentioned by these authors, we can find this proposal in the very first paper published for these problems. Kuby (1988) introduced the MaxSum, the MaxMin, and what this author called a multi-criteria approach, arguing that the MaxSum model is an appropriate way to choose among the many alternate optima of the MaxMin problem.

Parreño et al. (2021) perform a numerical and geometrical analysis of four diversity models: MaxMin, MaxSum, MaxMinSum, and MinDiff. Their analysis reveals that the MaxMin avoids very close elements but may select points either at a medium or at a large distance. On the other hand, the MaxSum favors the selection of points at a large distance but permits very close elements. Therefore, one of the conclusions of their study is that the combination of these two first models, in the way described above, would lead to a more robust model. The authors formulate this combined model, called the bi-level MaxSum problem, by introducing d^* as the optimal value of the MaxMin model (solved first), as follows:

$$\begin{array}{lll} \text{Maximize} & \sum_{i<j} d_{ij} x_i x_j & \\ \text{subject to:} & \sum_{i=1}^{n} x_i = m & \\ & d_{ij} \geq d^* x_i x_j & i, j = 1, \ldots, n. \\ & x_i \in \{0, 1\} & i = 1, \ldots, n. \end{array} \tag{11}$$

Fig. 2 shows the MaxMin optimal solution (left), the MaxSum optimal solution (center), and the Bi-level optimal solution (right), of an Euclidean instance with $n = 50$ elements from which we select $m = 5$.

The MaxMin optimal solution depicted in the left diagram of Fig. 2 shows the typical disposition of the solutions of this model identified by Parreño et al. (2021), in which the elements are scattered in the plane providing a disperse selection that may include the central region. A criticism of that selection, however, would be the point in the left part of the diagram, around coordinates (5,40), instead of which we could easily select a better one in terms of global dispersion. As a matter of fact, the MaxSum value of that solution is 829.8, which is relatively low compared with

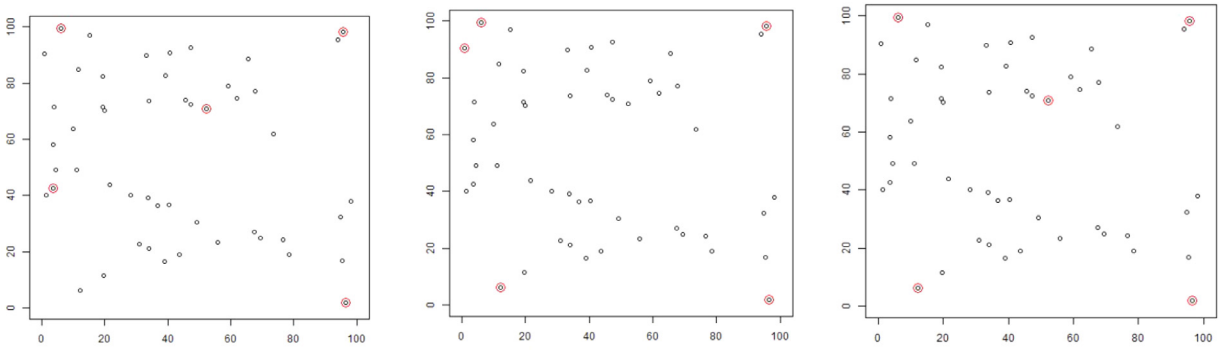

Fig. 2. MaxMin, MaxSum, and Bi-level optimal solutions.

the MaxSum optimal value of 942.8. The optimal MaxSum solution corresponding to that value is shown in the center diagram of the figure, and also has the typical disposition of that model, avoiding the central part and with the issue of selecting two points very close (see the upper left corner of the square). The diagram on the right clearly shows that the bi-level model provides an "in-between" solution, considering the optimal solutions of the two original models. Instead of the point around coordinates (5,40), it selects the point around coordinates (15,5). This "swap" does not change the MaxMin objective function, which is 51.4 in both models, but is able to increase the MaxSum value from 829.8 to 885.2 in the bi-level model.

A natural extension of the bi-level model is the bi-objective model, in which both objectives, MaxSum and MaxMin, are considered as equally important, and treated with the standard multi-objective methodology. Colmenar, Martí, & Duarte (2018) first adapt the standard solvers NSGA-II and SPEA, and then propose several metaheuristics to the bi-objective problem. In particular, the authors consider two construction-based methods, namely GRASP and Iterated Greedy, and two trajectory-based, namely tabu search and VNS. The comparison of the methods include the hypervolume, coverage, and epsilon indicator of the approximation of the Pareto front obtained with each method. The comparison shows that tabu search is able to obtain the best solutions.

4.3. The MaxMean model

Martí & Sandoya (2013) propose an advanced GRASP for the MaxMean problem introduced by Prokopyev et al. (2009) that they called the Equitable Dispersion problem, in which the number of selected elements is not set beforehand. In particular, the authors target general instances in which distances can take positive and negative values and do not necessarily satisfy the usual distance properties, such as the triangular inequality, reflecting for example the polarization that occurs when people get together in groups, in which we can identify clusters of individuals, with a high attraction within clusters and a high repulsion between clusters, and with no room for indifference. Note that the Max-Mean Dispersion Problem is polynomially solvable if all the distances are non-negative, but it is strongly NP-hard if they can take positive and negative values. The authors propose a GRASP constructive algorithm based on a non-standard combination of greediness and randomization, a local search strategy based on the variable neighborhood descent methodology, and a path relinking post-processing. This later method is based on a measure to control the diversity in the search process. The empirical comparison with a previous standard GRASP (Prokopyev et al., 2009) favors the proposed method.

The paper by Martí & Sandoya (2013) drew the attention of researchers working on diversity problems to the MaxMean model. A few years later, Carrasco et al. (2015) propose a tabu search based on constructive and destructive moves, and three local search methods with nested neighborhoods. Their tabu search algorithm, built upon short-term and long-term strategies, outperforms the previous GRASP methods. Della Croce, Garraffa, & Salassa (2016) propose a very interesting combination of methods in a 3-stage algorithm: a quadratic integer solver to find promising values for the number of selected elements to generate initial solutions, a local branching scheme, and a path relinking post-processing. Lai & Hao (2016) hybridize the tabu search methodology with an evolutionary method thus creating a memetic algorithm that improves upon previous methods according to their extensive computational comparison. As shown in the subsection on the MaxSum problem, this type of memetic algorithm has been already applied to other diversity models, and we can therefore conclude that it is a robust method that performs well across different models.

Brimberg, Mladenović, Todosijević, & Urošević (2019) propose a simple VNS for the MaxMean problem. The authors identify the minimum number of ingredients that makes a VNS based heuristic as simple and user friendly as possible, while at the same time achieving high-quality results. To clearly state this goal, the paper title starts with the expression *Less is more*, and the proposed algorithm follows the general variable neighborhood search methodology. The experimental comparison shows that, in spite of its simplicity, this VNS competes very well with the complex tabu search by Carrasco et al. (2015).

We end the revision on the MaxMean model with an exact algorithm. Garraffa, Della Croce, & Salassa (2017) consider the non-convex quadratic fractional formulation (see (5)) from which a semidefinite programming (SDP) relaxation can be derived. This relaxation is tightened by means of a cutting plane algorithm which iteratively adds the most violated inequalities. The proposed approach embeds the SDP relaxation and the cutting plane algorithm into a branch and bound framework. Computational experiments show that the proposed method is able to solve to optimality instances with up to 100 elements in less than 5 hours of CPU time.

Lai, Hao, & Glover (2020) adapted their memetic algorithm proposed for the MaxMean (Lai & Hao, 2016) to the Generalized MaxMean (see formulation (6) above), in which some weights multiply the objective function. This is the first heuristic for this extended model introduced in Prokopyev et al. (2009).

4.4. Other unconstrained diversity models

As mentioned in Section 3.3, Prokopyev et al. (2009) introduced in the previous period several diversity models that did not receive attention at that time. We have just reviewed above several contributions on the MaxMean model, and we are going to see now a few more on the **MaxMinSum** and **MinDiff** as well.

Building on the main ideas applied to different metaheuristics for the MaxSum and MaxMin models, Aringhieri, Cordone, & Grosso (2015) propose some constructive procedures and a Tabu Search algorithm for the MaxMinSum and MinDiff models. In

particular, the authors investigate the extension to this new context of key features such as initialization, tenure management and diversification mechanisms. The computational experiments show that the proposed algorithms perform effectively on the publicly available benchmarks. Martínez-Gavara, Campos, Laguna, & Martí (2017) integrate GRASP and Tabu Search in a scheme in which elements are selected and des-selected thus oscillating around the feasibility boundary defined by the problem constraint. The authors tested six different variants of GRASP, and three variants of the strategic oscillation. The final method is compared with a commercially available optimization software for combinatorial problems www.localsolver.com. Amirgaliyeva, Mladenović, Todosijević, & Urošević (2017) apply different variants of the variable neighborhood search methodology to the MaxMinSum, including the variable formulation search that iterates over different formulations to escape from local optima. The authors compare their method with the tabu search by Aringhieri et al. (2015), obtaining better results.

The most recent approach for the MaxMinSum is due to Lai, Yue, Hao, & Glover (2018), in which a solution-based tabu search is proposed. It is worth mentioning that the standard tabu search implementation is based on attributive memory, in which only key properties (called attributes) of moves or solutions are stored to avoid cycling. In this implementation however, the authors consider an interesting variant in which instead of an attribute, they record the entire solution by means of hash functions to speed up its management. An exhaustive empirical comparison with previous methods identifies this tabu search as the best method published so far for the MaxMinSum.

We consider now the **MinDiff model**, for which Duarte, Sánchez-Oro, Resende, Glover, & Martí (2015) proposed a GRASP with Exterior Path Relinking. Given two solutions, S and S', the standard implementation of the path relinking starts from the *initiating solution* S and gradually transforms it into the *guiding solution* S'. This transformation is accomplished by swapping out elements selected in S with elements in S', generating a set of *intermediate solutions*. The exterior Path Relinking introduces in the initiating solution characteristics not present in the guiding solution with diversification purposes. Specifically, it removes from the initiating solution those elements which also belong to the guiding solution, obtaining intermediate solutions which are further away from both the initiating and the guiding solutions. The authors show that this method is able to obtain high quality solutions by comparing them with the optimal values obtained with CPLEX.

After Duarte's GRASP with Exterior Path Relinking, three heuristics have been proposed. They are basically adaptations of methods proposed for other diversity models to target the specific characteristics of the MinDiff model. In particular, Mladenović, Todosijević, & Urošević (2016) propose a VNS, Zhou & Hao (2017) an iterated local search, and Lai, Hao, Yue, & Gao (2019) a solution-based tabu search, which according to their computational testing, is currently the state-of-the-art method for this problem.

A major criticism of the two models reviewed in this subsection is its lack of practical significance. Parreño et al. (2021) analyze these two models, in connection with the rest of diversity models. The first conclusion of their study is that the MaxSum and MaxMinSum provide similar solutions, and considering the relatively large amount of research already done in the MaxSum model, it is not well justified the need of the recently introduced MaxMinSum one (especially because it is more complicated). In particular, their empirical analysis reveals that the optimal solution obtained with one model scores very well in the other model, presenting a small deviation with respect to its optimum (0.8% on average on the MDPLIB). Additionally, both models present an average correlation of 0.74, and in many cases it is larger than 0.9. Regarding the geometrical disposition of its solutions, they select points close to the borders of the space, and with no points in the central region. Fig. 3 shows the MaxMinSum optimal solution (left), and the MaxSum optimal solution (right), of a Euclidean instance with $n = 100$ elements from which we select $m = 20$. It is clear that both solutions are very similar (they only differ in one point).

Regarding the MinDiff, Parreño et al. (2021) also recommend to avoid the use of this model in its current formulation. Their analysis reveals that it seeks for inter-distance equality among the selected points, but ignores how large or small these distances are. This model balances the selection of points, achieving equity in this way; however, it seems difficult to justify the selection of balanced points at a very small distance, as shown in the example of Fig. 4 with $n = 25$ elements from which we select $m = 3$.

To sum it up, it seems that researchers have focused their attention on these two problems as a way to evaluate complex metaheuristics, but without considering their true practical significance. More research is needed to conclude if they are artificial problems or require a better formulation to capture diversity and equity in a more realistic way.

4.5. Constrained dispersion models

As mentioned above, in the previous decade Rosenkrantz et al. (2000) introduced several diversity models constrained in terms of cost and capacity, motivated by their practical applications in facility location. In these last few years, several models have been developed from this seminal paper.

Peiró, Jiménez, Laguardia, & Martí (2021) considered the model of maximizing the diversity subject to capacity constraints. This model, as stated in Rosenkrantz et al. (2000), is built upon the MaxMin, by replacing the typical cardinality constraint with capacity constraints. The authors called it the **Capacitated Dispersion Problem** (CDP), and proposed a hybridization of GRASP and VND implemented within the Strategic Oscillation framework. A straightforward formulation, based on the standard binary variables x_i, a capacity value c_i for each node i, and a capacity threshold B indicating the desired level of service, follows:

$$\begin{aligned} &\text{Maximize} && Min_{i,j\in M} \quad d_{ij} \\ &\text{subject to:} && \sum_{i=1}^{n} c_i x_i \geq B \\ & && M = \{i \in V : x_i = 1\} \\ & && x_i \in \{0,1\} \qquad\qquad i = 1,\ldots,n. \end{aligned} \tag{12}$$

Martí, Martínez-Gavara, & Sánchez-Oro (2021) propose a mathematical model and a heuristic based on the Scatter Search methodology to maximize the diversity while satisfying the capacity constraint in the CDP. Their heuristic algorithm outperforms the previous heuristic on the 100 instances tested, and the model is able to solve the medium size instances in this set to optimality. In particular, the authors adapt the exact method by Sayyady & Fathi (2016) for the MaxMin to the CDP. It basically solves iteratively the node packing problem, which finds a maximum cardinality subset of nodes in an auxiliary graph, so that no two nodes in this subset are adjacent to each other. With a time limit of 3600 seconds, Gurobi is able to solve all the instances with $n = 50$, and $n = 150$, and some of the instances with $n = 500$.

This same year in which we are writing this paper, 2021, a new constrained model has been published. Martínez-Gavara et al. (2021) consider the model in which capacity and cost constraints are included. This model was labeled as Max-Dist/Cap/Cost by Rosenkrantz et al. (2000), and it is coined now as the **Generalized Dispersion Problem** (GDP). It basically adds a cost constraint to the CDP. For each element i, it considers an associated cost, a_i, and a maximum budget K that cannot be exceeded. Martínez-Gavara et al. (2021) also propose another model that includes both fixed and variable costs, to model in a more realistic way some location

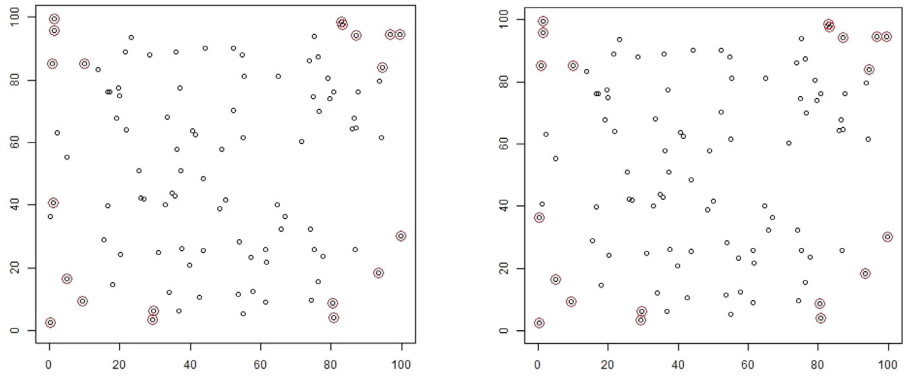

Fig. 3. MaxMinSum (left) and MaxSum (right) optimal solutions.

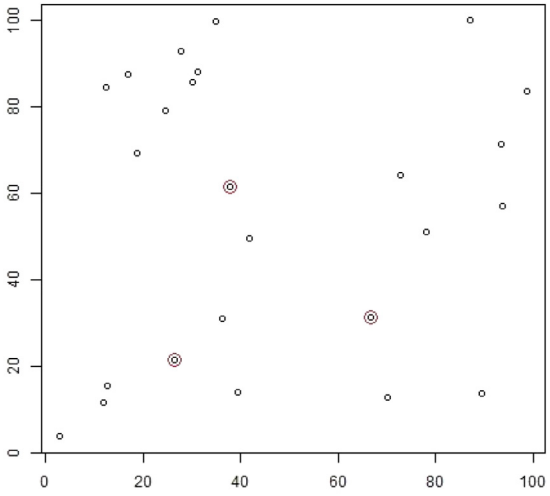

Fig. 4. MinDiff optimal solution.

problems.

$$
\begin{array}{lll}
\text{Maximize} & m & \\
\text{subject to:} & \sum_{i=1}^{n} c_i x_i \geq B & \\
& \sum_{i=1}^{n} a_i x_i \leq K & \\
& m \leq d_{ij} + D(1-x_i) + D(1-x_j) & i,j = 1,\ldots,n : j > i. \\
& x_i \in \{0,1\} & i = 1,\ldots,n.
\end{array}
\tag{13}
$$

It is noteworthy the relative relationship between these constrained models and the well-known discrete p-median problem. In both models we want to select some locations to establish some facilities; however, the p-median solution assigns each client to a facility, which is not the case of dispersion problems. In general terms, we may say that p-median models emphasize the distance between facility and clients, while dispersion models emphasize the distance among facilities.

Martínez-Gavara et al. (2021) illustrate the practical use of this model with the location problem of a medical corporation that wants to set several facilities, such as clinics or hospitals, in a certain territory. In this context, the set of nodes would represent the potential locations for the facilities (such as hospitals or clinics), the capacity value B the minimum number of patients that they want to attend, and the cost limit K their budget. Maximizing the inter-distance between facilities translates the objective of scatter the clinics over the territory to cover it, in a similar way that the p-median minimizes the distance between the facility and the assigned patients. Note however, that in this model, we are not assigning the clients (patients) to clinics, and we are giving them the freedom to select the one that they prefer, which is precisely what many medical corporations do.

5. The MDPLIB library of benchmark instances

The benchmark instances for the diversity problem come from different sources that have been added over the years. The most used library is the MDPLIB; however, other instances have also been considered, such as the OR-Lib. On the other hand, the library for the constrained dispersion problems is quite recent, and since it is derived from the MDPLIB, we propose to include all of them in an extended version of the MDPLIB, called MDPLIB 2.0. A detailed description of the different sets of instances follows.

The original MDPLIB collects a total of 315 instances available at www.uv.es/rmarti/paper/mdp.html with a mirror server in www.optsicom.es/mdp. Martí et al. (2010) compiled ten years ago this comprehensive set of benchmark instances representative of the collections used for computational experiments in the MDP. The library contains three sets of instances collected from different papers and named after their authors: GKD (Glover, Kuo, and Dhir), MDG (Martí, Duarte, and Gallego), and SOM (Silva, Ochi, and Martins). All the instances were randomly generated. The generators were not built according to any specific application, but they were designed with the purpose of being a challenge for heuristic methods, mainly on the MaxSum problem. However, these instances have been extensively used in all the diversity models proposed, and some studies point out that not all of them are appropriate for some models.

In this section, we first describe in detail each set of instances, which contains different subsets according to their source. We consider three sets of instances depending on the type of values in their distance matrices: Euclidean, Real, and Integer. In our descriptions below, we analyze these sets, and propose some changes to update the library. We will refer to the new library as MDPLIB 2.0.

1. **Euclidean instances set.** This data set consists of 215 matrices for which the values were calculated as the Euclidean distances from randomly generated points with coordinates in the 0 to 10 range. It collects four subsets, namely GKD-a, GKD-b, GKD-c, and GKD-d:
   (a) GKD-a: Glover et al. (1998) introduced these 75 instances in which the number of coordinates for each point is generated randomly in the 2 to 21 range. The instance sizes are such that for $n = 10$, $m = 2, 3, 4, 6$ and 8; for $n = 15$, $m = 3, 4, 6, 9$ and 12; and for $n = 30$, $m = 6, 9, 12, 18$ and 24.
   (b) GKD-b: Martí et al. (2010) generated these 50 matrices for which the number of coordinates for each point is generated randomly in the 2 to 21 range and the instance sizes are

such that for $n = 25$, $m = 2$ and 7; for $n = 50$, $m = 5$ and 15; for $n = 100$, $m = 10$ and 30; for $n = 125$, $m = 12$ and 37; and for $n = 150$, $m = 15$ and 45.

(c) GKD-c: Duarte & Martí (2007) generated these 20 matrices with 10 coordinates for each point and n = 500 and m = 50.

(d) GKD-d: Parreño et al. (2021) generated 70 matrices for which the values were calculated as the Euclidean distances from randomly generated points with two coordinates in the 0 to 100 range. For each value of $n = 25, 50, 100, 250, 500, 1000$, and 2000, they considered 10 instances with $m = \lceil n/10 \rceil$ and 10 instances with $m = 2\lceil n/10 \rceil$, totalizing 140 instances. The main motivation of this new set is to include the original coordinates in the instances files that unfortunately are not publicly available nowadays for the other subsets. In this way, researchers may represent the solutions in line with the work in Parreño et al. (2021).

We replace the original sets GKD-a and GKD-b in the benchmark library with the new set GKD-d, in which the instances are generated in the same way but their corresponding files contain the coordinates. Note that the new set contains very large instances not considered in the original sets.

2. **Real instances set.** This data set consists of 140 matrices with real numbers randomly selected according to a uniform distribution.

(a) MDG-a. This data set contains 60 instances. Duarte & Martí (2007) generated 40 matrices with real numbers randomly selected in $[0, 10]$ and called them *Random Type I instances*, 20 of them with $n = 500$ and $m = 50$, and the other 20 with $n = 2000$ and $m = 200$. Parreño et al. (2021) generated 20 additional matrices with $n = 100$ and real numbers randomly selected in $[0, 10]$ that can be solved to optimality.

(b) MDG-b. This data set contains 60 instances. Originally, Duarte & Martí (2007) created this set with 40 matrices generated with real numbers randomly selected in [0, 1000] and called them *Random Type II instances*. 20 of them have $n = 500$ and $m = 50$, and the other 20 have $n = 2000$ and $m = 200$. Parreño et al. (2021) generated 20 additional matrices with $n = 100$ and real numbers randomly selected in [0, 1000],

(c) MDG-c. Considering that many heuristics were able to match the best-known results in many of the instances previously introduced, Martí et al. (2013) proposed this data set with very large instances in 2013. It consists of 20 matrices with randomly generated numbers according to a uniform distribution in the range [0, 1000], and with $n = 3000$ and $m = 300$, 400, 500 and 600.

3. **Integer instances set.** This data set consists of 170 instances where the distance matrices are integer random numbers generated from an integer uniform distribution.

(a) ORLIB: This is a set of 10 instances with $n = 2500$ and $m = 1000$ that were proposed for binary problems (Beasley, 1990). The distances are integers generated at random in $[-100, 100]$ where the diagonal distances are ignored.

(b) PI: Palubeckis (2007) generated 10 instances where the distances are integers from a [0,100] uniform distribution. 5 of them are generated with $n = 3000$ and $m = 0.5n$, and 5 with $n = 5000$ and $m = 0.5n$. The density of the distance matrix is 10%, 30%, 50%, 80% and 100%.

(c) SOM-a. These 50 instances were generated by Martí et al. (2010) with a generator developed by Silva et al. (2004) with integer random numbers between 0 and 9 generated from an integer uniform distribution. The instance sizes are such that for $n = 25$, $m = 2$ and 7; for $n = 50$, $m = 5$ and 15; for $n = 100$, $m = 10$ and 30; for $n = 125$, $m = 12$ and 37; and for $n = 150$, $m = 15$ and 45.

(d) SOM-b. These 20 instances were generated by Silva et al. (2004) with the same random generator from SOM-a. The instance sizes are such that for $n = 100$, $m = 10, 20, 30$ and 40; for $n = 200$, $m = 20, 40, 60$ and 80; for $n = 300$, $m = 30, 60, 90$ and 120; for $n = 400$, $m = 40, 80, 120$, and 160; and for $n = 500$, $m = 50, 100, 150$ and 200.

(e) MGPO: To complement the sets above, we consider 80 large matrices with relatively low m values. Specifically, we generate 40 instances with $n = 1000$ and integer numbers randomly selected in $[1, 100]$, 20 of them with $m = 50$ and 20 with $m = 100$. Similarly, we generate 40 matrices with $n = 2000$ and integer numbers randomly selected in $[1, 100]$, 20 of them with $m = 50$, and 20 with $m = 100$.

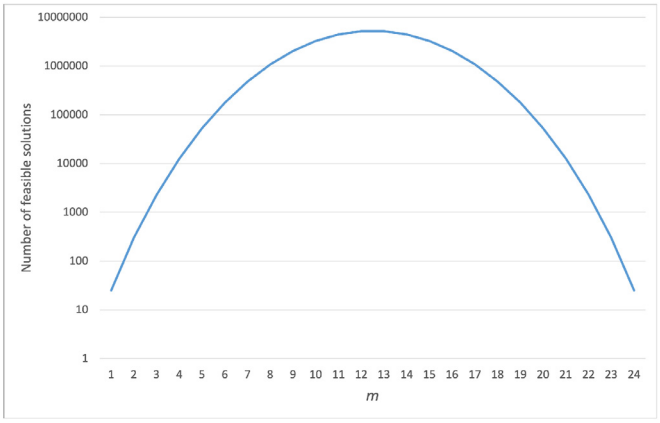

Fig. 5. Number of solutions of an instance with $n = 25$.

A final note on the use of instances is its applicability to the different models. It must be noted that some of them were introduced for the MaxSum model, and could not be adequate for other diversity models. This is especially true in the case of some instances in the SOM set that contain so many 0 values that all feasible solutions have a minimum distance value of 0. Our empirical analysis in Section 6 shows that 23 instances in the SOM set have an optimal MaxMin value of 0, and therefore if we apply a heuristic and obtain a solution with a value of 0 in the MaxMin objective, this is not a reliable measure of its assessment. Researchers have to be very careful when using this set to test other models than the classic MaxSum. We are including a note in the MDPLIB 2.0 identifying these 23 instances.

A simple but important argument when considering an instance to compare methods is its difficulty based on the ratio between the total number of elements n, and the number of them to be selected, m. Since any selection of m elements is a solution, the number of feasible solutions is simply $C_m^n = \frac{n!}{m!(n-m)!}$. Therefore, for a given value of n, the closer m is to $n/2$, the more difficult the instance is. For example, an instance with $n = 25$ and $m = 2$ only has 300 solutions, while an instance with $n = 25$ and $m = 10$ has more than 3 million solutions. Fig. 5 shows the number of solutions as a function of m for an instance with $n = 25$.

5.1. Constrained benchmark instances

The benchmark set of instances in the constrained dispersion problem is derived from the MDPLIB described above. Specifically, Peiró et al. (2021) and Martínez-Gavara et al. (2021) select a subset of 50 instances to generate the new benchmark set. It consists of 30 instances from GKD set, 10 of each size ($n = 50$, $n = 150$, and

Table 5
MDPLIB 2.0 benchmark library.

Set	# Instances	Type	Range of n	Range of m
GKD-c	20	Euclidean	500	50
GKD-d	140		[25, 2000]	[3, 400]
MDG-a	60		[100, 2000]	[50, 200]
MDG-b	60	Real numbers	[100, 2000]	[50, 200]
MDG-c	20		3000	[300, 600]
ORLIB	10	Integer	2500	1000
PI	10	numbers	{3000, 5000}	{1500, 2500}
SOM-a	50		[25, 150]	[2, 45]
SOM-b	20		[100, 500]	[10, 200]
MGPO	80		[1000, 2000]	[50, 100]
Const - (CDP)	100	Constrained	[50, 500]	–
Const - (GDP)	200	Constrained	[50, 500]	–
Total	770		[25, 5000]	[2, 2500]

$n = 500$), 10 instances of size 500 from the MDG set, and finally, 10 more instances of size 50 are selected from the SOM set.

The capacity of each node i, c_i in Eq. (12), is randomly generated with a uniform distribution between $[1, 1000]$ for each of these original instances. Then, the minimum capacity B is computed as the total capacity multiplied by 0.2 or 0.3, thus two instances are created for each of these 50 instances. So, the Const.-(CDP) benchmark contains 100 instances. Moreover, in the GDP, for each of these 100 instances, the cost a_i of a node i, see Eq. (13), is generated by a uniform distribution between the values $c_i/2$ and $2\,c_i$. As in the capacity constraint, the maximum budget K is computed as the sum of all the costs values multiplied by a factor between 0.2 and 0.3. Therefore, in the Const.-(GDP) benchmark, each original instance in the MDPLIB produces 4 instances, thus obtaining a set of 200 instances.

We have generated an additional set of large instances. In particular, we consider 20 new instances in each set: 20 Euclidean (GKD-d) with $n = 2000$, 20 Real (MDG-c) with $n = 3000$, and 20 Integer (MGPO) with $n = 2000$. The capacity and cost values are generated as described above.

We finish the description of the instances, summarizing the new library, MDPLIB 2.0, in Table 5. This table shows the number of instances, type, and the range of n and m in each subset. In general terms, Euclidean instances are based on location problems, Real instances were generated to pose a challenge for heuristics, and Integer instances somehow are related to rankings and preferences. However, none of them are directly based on applications. A major criticism of most of these instances is their lack of connection with real problems. In our opinion they should be closely linked to real applications. In fact, related fields, such as location or grouping, have well-known data sets based on important applications, which constitutes one of the foundations of Operations Research. We would suggest the study of real problems to generate future benchmarks.

6. Computational experiments

In this section we address the two diversity problems that have been extensively studied, the MaxSum and MaxMin. Considering that the number of methods proposed for them is very large and, in many cases, the comparisons performed are partial, with just a few methods and a fraction of the instances described in Section 5, we perform a complete comparison to clearly established the state-of-the-art methods for these two problems. We would like to thank the authors who kindly made their codes available to us. All the experiments are conducted on a computer with a 2.8 GHz Intel 369 Core i7 processor with 16 GB of RAM.

6.1. The MaxSum model

Martí et al. (2013) presented an extensive computational experimentation to compare 10 heuristics and 20 metaheuristics for the MaxSum problem (see Table 1). This comparison reveals that, the first heuristics proposed in the early period, C2 and D2, perform very well considering their simplicity, and in the set of complex metaheuristics proposed in the expansion period, B-VNS (Brimberg et al., 2009) and ITS (Palubeckis, 2007) exhibit the best results (see Table 2). Since then, several new efficient methods have been published (see Section 4), being the Memetic Evolution Strategy MSES (De Freitas et al., 2014), the Memetic Tabu Search TS-MA (Wang et al., 2014) and the opposition-based memetic algorithm OBMA (Zhou & Hao, 2017) the most recent ones. We consider these seven methods and the solutions obtained with CPLEX in our comparison.

In line with the previous comparisons previously published, we consider two time horizons in our testing: 10 seconds and 600 seconds of CPU time. In our first experiment, we exclude the MSES (De Freitas et al., 2014) because we are running its Matlab code provided by the authors that requires much more than the 10 seconds considered in this experiment. Table 6 reports the results of the other six heuristics referenced above run for 10 seconds. It also reports the solutions of the CPLEX solver with mathematical model (4) described above run for 1 hour. Note that in many cases CPLEX is not able to certify the optimality, and we report its best feasible solution found (current lower bound when the time limit expires). This table shows the average percentage deviation from the best solution known (% *dev*), and the number of best solutions found (# *best*). Results are reported for each instance set. In the case of CPLEX, % *dev* is only reported in a set, when it obtains feasible solutions in all the instances in that set.

Table 6 shows that, as expected, metaheuristics obtain better results than simple heuristics. In particular, the most recent published method, OBMA, obtains the best results overall, with an average percentage deviation of 0.16% and 327 best solutions found in the experiment. Note that TS-MA is able to slightly improve OBMA in terms of the average percentage deviation; however, a p-value < 0.001 of the one-sided pairwise Wilcoxon test confirms the superiority of OBMA. On the other hand, this table also shows that most of the problems are too large to be solved with CPLEX, and only in some of the instances sets it obtains feasible solutions.

If we compare the best method proposed in each period, we can see that in the early period, the best results were obtained with D2 that presents an average deviation of 36.95%. In the expansion period (second decade in our study), the best method is B-VNS, and the percentage deviation drops to 0.2%. Finally, in the development period (last decade) a slight improvement is achieved with very complex methods, being OBMA the best method (closely followed by TS-MA), with a deviation of 0.16 (and 0.02 for TS-MA).

In the next experiment, we compare the best methods identified for each period time, namely D2, B-VNS, and OBMA, run with a time limit of 600 seconds per instance. We include in this experiment the solutions obtained with CPLEX and MSES which require on average about an hour of CPU time. Table 7 shows the same statistical parameters than the previous table. The results in this table show that simple heuristics are not able to improve complex metaheuristics over a long period of time, and OBMA emerges as the best algorithm again, obtaining the best percentage deviation overall. Furthermore, OBMA exhibits a remarkable 99% of the best solutions, while this percentage in the B-VNS is around 70%. The pairwise Wilcoxon statistical test confirms that OBMA outperforms B-VNS, with a p-value less than 0.001. These comments are in line with the results in the previous experiment.

The last experiment in this subsection evaluates how close the solutions of the algorithms are with respect to the optimal values.

Table 6
Comparison of the best methods for the Max-Sum problem in 10 seconds.

	Instance class								
	GKD-c	GKD-d	MDG-a	MDG-b	MDG-c	SOM-a	SOM-b	SOM-c	all
# inst.	20	140	60	60	20	50	20	80	450
% dev									
CPLEX	3.83	3.05	–	–	–	2.45	6.40	–	-
C2	97.35	16.54	74.51	74.04	99.56	85.65	96.45	19.05	70.39
D2	22.27	41.33	40.59	28.27	76.23	41.01	22.73	23.21	36.95
B-VNS	0.00	0.06	1.18	0.07	0.08	0.00	0.00	0.20	0.20
ITS	0.00	0.58	1.19	0.10	0.20	0.05	0.00	0.25	0.30
TS-MA	0.02	0.06	0.01	0.04	0.04	0.00	0.00	0.02	0.02
OBMA	0.00	0.06	1.14	0.03	0.00	0.00	0.00	0.03	0.16
# best									
CPLEX	0	46	0	0	0	18	0	0	65
C2	0	0	0	0	0	0	0	0	0
D2	0	0	0	0	0	0	0	0	0
B-VNS	19	108	22	0	0	50	20	13	232
ITS	19	109	20	24	0	48	19	10	249
TS-MA	1	45	58	44	0	50	20	63	281
OBMA	19	108	28	24	20	50	20	58	327

0.00 means less than 0.001.

Table 7
Comparison of the best methods for the Max-Sum problem in 600 seconds.

	Instance class								
	GKD-c	GKD-d	MDG-a	MDG-b	MDG-c	SOM-a	SOM-b	SOM-c	all
# inst.	20	140	60	60	20	50	20	80	450
% dev									
CPLEX	3.83	2.98	–	–	0.00	2.45	6.40	–	–
D2	10.05	24.99	20.03	18.44	76.23	26.92	19.26	19.48	29.92
B-VNS	0.00	0.00	0.01	0.01	0.02	0.00	0.00	0.05	0.01
MSES	0.00	0.71	1.15	0.72	0.41	0.00	0.07	0.89	0.49
OBMA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
# best									
CPLEX	0	79	3	5	0	18	0	0	105
D2	0	0	0	0	0	0	0	0	0
B-VNS	20	138	43	0	2	50	20	45	318
MSES	20	126	17	12	0	50	9	0	244
OBMA	20	138	60	60	20	50	20	80	447

0.00 means less than 0.001.

Table 8
Comparison with 45 optimal values obtained with CPLEX in the MaxSum.

Procedure	C2	D2	B-VNS	ITS	OBMA	TS-MA	MSES
% gap	33.54	25.03	0.00	0.05	0.00	0.00	0.00
# opt	0	0	45	44	45	44	45

0.00 means less than 0.001.

We can compute it for the 45 small instances that CPLEX is able to optimally solve. Table 8 shows the average percentage deviation from the optimal solution (*% gap*) and the number of optimal solutions found by each algorithm (*# opt*) over the set of these 45 instances. Since the size of these instances is small ($n \in \{25, 50\}$), and the number of elements to be selected is less than 7 ($< n/3$), we may consider these 45 instances as easy to solve. However, simple heuristics, such as C2 and D2, are not able to match the optimal solutions, while metaheuristics can achieve almost all of them. Furthermore, the results obtained by the heuristics are on average less than 34.0% away from the optimal value.

To summarize the situation on the MaxSum problem, we conclude that simple heuristics obtain low quality solutions, and we should avoid their use. The efforts made in the last two decades on this problem, result in very efficient metaheuristics that are able to obtain good solutions even in very short running times, such as the 10 seconds tested. Results obtained with the metaheuristics in the development period (last decade analyzed) slightly improve those in the previous period, and many of them would be adequate for a large range of applications in which a medium size instance has to be solved. Regarding the optimal values, the MaxSum model implemented in CPLEX is only able to certify optimal solutions in a small fraction of the instances (around a 10% overall), which indicates that this model is still a challenge for the operation research community, and further research is necessary to obtain a model that could increase the number of optimal solutions found.

6.2. The MaxMin model

This section describes the numerical experiments that we have performed to test the efficiency of the most representative algorithms for the MaxMin model. The first two algorithms that we include in the comparison belong to the early period, and fall under the category of heuristic algorithms. Specifically, we adapt the constructive and destructive algorithms proposed by Glover et al. (1998) to the MaxMin problem, and we name them as C2Ad and D2Ad, respectively. They are similar to those proposed by Erkut (1990). At the end of the expansion period, Resende et al. (2010) performed a numerical analysis to compare their proposed algorithm GPR with the previous metaheuristics, and conclude that GPR outperformed the state-of-art at that time (see Table 3). So, we consider GPR in the next comparison as the representative algorithm of that period. Finally, in the development period (the last decade in our study), we can only find the metaheuristic proposed

Table 9
Comparison of the best methods for the Max-Min problem in 10 seconds.

	Instance class								
# inst.	GKD-c 20	GKD-d 140	MDG-a 60	MDG-b 60	MDG-c 20	SOM-a 50	SOM-b 20	SOM-c 80	all 450
% dev									
CPLEX	4.55	0.22	–	–	–	0.00	15.00	–	–
C2Ad	56.06	91.35	65.38	98.16	100.00	68.61	35.00	100.00	76.82
D2Ad	16.01	42.41	44.51	74.54	75.23	62.53	35.00	86.09	54.54
GPR	4.00	30.46	7.87	54.02	100.00	7.30	10.00	64.21	34.73
DropAdd-TS	0.01	21.00	1.35	25.43	0.00	2.72	0.00	0.00	6.31
# best									
CPLEX	2	139	40	20	0	50	17	0	268
C2Ad	0	0	20	0	0	15	13	0	48
D2Ad	0	0	20	0	0	12	13	0	45
GPR	0	29	39	14	0	44	18	0	144
DropAdd-TS	20	32	58	39	20	47	20	80	316

0.00 means less than 0.001.

Table 10
Comparison with 227 optimal values in the MaxMin model.

Procedure	C2Ad	D2Ad	DropAdd-TS	GPR
% gap	88.84	52.03	18.76	23.08
# opt	28	25	114	107

by Porumbel et al. (2011), which consists in combining add and drop operations with a simple tabu search (named DropAdd-TS).

In contrast to what happens with the MaxSum model, in the last decade, new formulations have been proposed to the MaxMin, increasing the number of optimal solutions that can be solved with CPLEX. Results in Tables 9 and 10 are obtained with the model proposed by Sayyady & Fathi (2016), running it with a time limit of 1 hour per instance.

As in the previous section, we first compare the results obtained with the four algorithms run with a small time limit (10 seconds), including in the comparison the CPLEX results. Table 9 summarizes the results by instance set, and shows the average percentage deviation from the best solution known (*% dev*), and the number of best solutions found (*# best*). As in the previous section, the average percentage deviation for CPLEX is only reported in a set, when it obtains feasible solutions in all the instances in that set.

As expected, Table 9 shows that metaheuristics outperform heuristics, and DropAdd-TS arises as the best algorithm overall, with an average percentage deviation of 6.31% and 316 best solution found in the experiment. It is worth mentioning that CPLEX, with the Sayyady & Fathi (2016) formulation, is able to obtain a total of 268 bests solutions out of 450 in the experiment (around 60% overall), even improving the results achieve by GPR. This formulation solves to optimality many instances of large size (with $n = 1000$), and is able to obtain high quality lower bounds in even larger instances ($n = 2000$). Finally, comparing the two simple heuristics considered, we can see that the destructive method D2Ad obtains better solutions than the constructive one (C2Ad). Specifically, D2Ad presents an average deviation of 54.51% in contrast to the average deviation of 76.82% that C2Ad obtains.

We repeat the same experiment performed above with a time horizon of 600 seconds. The results obtained are similar to those presented in Table 9 for 10 seconds, so we do not include the results here. It must be emphasized that GPR is able to decrease by 10% the percentage deviation to the best solution found in this experiment, and to increase its number of bests solution (*# best*) from 144 to 159. This makes sense since the methodology applied in this algorithm usually requires longer running times due to the combination of solutions.

Finally, the last experiment in this section has the objective to evaluate how far the solutions provided by the algorithms are from optima, or if they are able to match them. As in the previous section, we compare the four algorithms in the subset of instances that CPLEX optimally solves. In particular, the MaxMin model implemented in CPLEX is able to certify optimal solutions in 227 instances out of 450 (around 50%). Clearly, the new formulations that have been recently proposed for the MaxMin model allow to optimally solve instances with large size ($n \leq 1000$ in our benchmark set) with relatively low running times, as opposite to what happens in the MaxSum model. Table 10 shows the average percentage deviation from the optimal solution (*% gap*) and the number of optimal solutions found by each algorithm (*# opt*) over the set of these 227 instances. None of them is able to compete with the results obtained by CPLEX, although it must be noted that they require smaller running times.

6.3. The bi-level MaxSum model

As mentioned, Porumbel et al. (2011) proposed a combined model between the MaxMin and the MaxSum problems. They considered the MaxMin objective function, subject to the MaxSum as a secondary objective, based on the fact that there is a large number of optimal solutions for the MaxMin, so we look for the best one among them in terms of the MaxSum objective. Parreño et al. (2021) support this point with a geometrical argument since they disclose that the MaxMin avoids the selection of very close elements but can be at medium distances (not very far away from each other), while the MaxSum favors the selection of points at a large distance but permits very close elements, so in a way they complement each other. The authors called it the Bi-level MaxSum problem.

Porumbel et al. (2011) designed a tabu search heuristic, DropAdd-TS, specifically for this problem, in which the method tries to maximize both objectives (being the MaxSum secondary). Since we consider the bi-level model as a very interesting one, we perform an experiment to evaluate how good this algorithm is in maximizing the sum of distances over the set of optimal solutions of the MaxMin, and at the same time the practical significance of the model. Note that, since we are applying heuristics, we cannot guarantee the optimality, and therefore what we do to evaluate the quality of this method, is to compare it with a previous heuristic. In particular, we run the GRASP with Path Relinking, GPR by Resende et al. (2010) and the DropAdd-TS to solve our benchmark set of instances. Although GPR only minimizes the MaxMin, we evaluate both objectives, MaxMin and MaxSum, in its output solution. We do the same for the output of the DropAdd-TS.

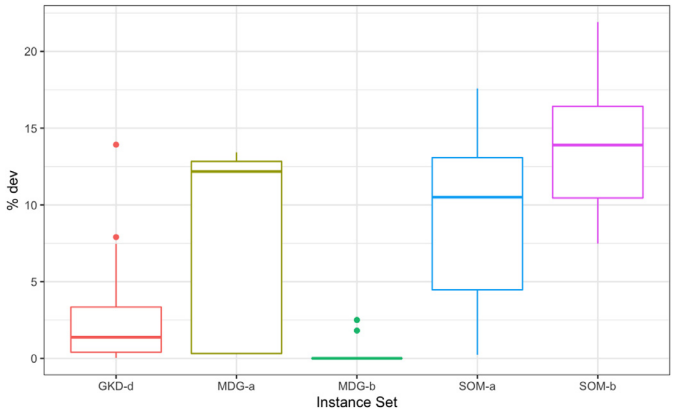

Fig. 6. MaxSum percentage improvement of DropAdd-TS with respect to GPR.

To perform a fair test about the ability of the DropAdd-TS to find good solutions in terms of the MaxSum, we only consider the instances in which both methods obtain the same value of the MaxMin objective. Fig. 6 shows the percentage improvement (*% dev*) of the DropAdd-TS MaxSum value with respect to the GPR MaxSum value. This figure shows a boxplot of the average percentage deviations in each instance set. Their positive values indicate that DropAdd-TS always obtains a better (larger) sum of distances than GPR in all the instances in which both methods obtain the same MaxMin value. This confirms that the Bi-level model permits to discriminate among solutions, selecting the best one overall. It also quantifies the relative contribution of the DropAdd-TS algorithm with respect to the GPR, thus certifying its superiority. We believe that this new model brings new research opportunities, since it clearly deserves to be further studied.

7. Conclusions

In **the early period** (1980–2000) two mathematical models were proposed to capture the notion of diversity, the MaxSum and MaxMin, and simple heuristics were applied to solve these models in short computational times. On the other hand, in this decade only small instances were considered. In the following decade, that we called **the expansion period**, the three main open problems at that time were approached. In particular, researchers consider other models to include different aspects of diversity, they introduce larger instances that pose a challenge to simple heuristics, and apply complex metaheuristics to efficiently solve the problems.

During the last decade, called **the development period** (2010 - to now), researchers have been mainly working on the lines proposed in the previous decade (described in Section 3). This is why we call it the development period because it intensifies the research over the known models (collected in Table 4), without proposing new ones. Authors limit themselves to the strict competition among methods, without extending the boundaries that currently define the field. We want to give credit to them because the competition among methods is now very hard, and the proposed methods both exact and heuristics are very sophisticated, but we believe that there is still some work to do on expanding the area. In the same way that heuristic methods require intensification and diversification for an efficient exploration of the solution space, we believe that the scientific methodology requires to revisit the models and problems to improve solving methods, but also to propose and explore new models to approach in a more realistic way the complexity of real problems, connecting in this way the area with related fields of knowledge.

Considering the characteristics of the solutions obtained by the different models, the most important conclusions are:

- The MaxSum and MaxMinSum provide similar solutions in terms of their geometrical location, since they select points close to the borders of the space, and with no points in the central region. Thus, it seems quite artificial the use of this latter complicated model. These models may select a few elements that are very close to each other. They reflect what we usually understand as **dispersion**.
- The MaxMin model generates solutions with a different structure than the MaxSum. It usually obtains equidistant points, and it does not avoid to select points in the central part. This mathematical formulation induces **representativeness**, more than dispersion.
- The MinDiff only seeks for inter-distance equality (**equity**) among the selected points, and ignores how large or small these distances are, thus neglecting diversity or dispersion, which could be an issue in many contexts.

7.1. Open problems

We finish our review pointing to potential new areas that, in our opinion, deserve the attention of researchers.

It is shown in this review that for every diversity model, many heuristics have been proposed, but only a **few exact methods**, if any. In spite of being the most studied model, we can only solve to optimality medium size instances for the MaxSum. The study of valid inequalities to strength mathematical models is nowadays a well established technique; however, it has not been applied to diversity models yet, with the exception of the MaxMin (with excellent results). The adaptation of these techniques to the diversity problems, including the polyhedral study of their feasible regions, may lead to significant progress in this field. On the other hand, considering that the MaxMin exact methods are very efficient, the challenge is now to design powerful metaheuristics that can obtain the already known optimal solutions in short running times.

In the last few years, **constrained models** have emerged as a natural extension of the classic ones to adapt diversity to real situations. Cost or capacity, that are common elements in many other location models, have been largely ignored in diversity models. In our opinion, their study in this context has just started, and we will witness important developments in these lines.

The two **equity models** proposed so far, MaxMinSum and MinDiff, present drawbacks that discourage their use as they are formulated now. However, we believe that the concept of equity may find its realm in Operations Research, but only requires to be better formulated. As a matter of fact, in facility location problems, there is a vast literature of equity measures. The bi-level formulation, recently considered for the MaxMin, may well be a good way to overcome the limitations of their initial formulations.

Although many papers on diversity mention some **applications**, such as biological preservation or obnoxious location, they usually do not elaborate on them. In fact, early papers pay more attention to describe the applications and connect solving methods with real problems (see for example Glover et al., 1995), but now most of the research in the field concentrates on the comparison among heuristics, solving instances artificially generated to pose a challenge to them. In line with the Operations Research perspective, we would suggest to connect models and solving methods with the applications that originated them, and to incorporate into the models and instances the specific characteristics of the applications. In our view, that would create many research opportunities, and what is more important, would transfer knowledge between theoretical research to real-life problems.

Acknowledgments

We are in debt to many people who contributed to diversity problems, since we build our work from theirs. The list of names in the references below give them somehow credit for their achievements. Additionally, we would like to thank to some friends and colleagues that significantly contributed to the field: Mike Kuby; Fred Glover and Manuel Laguna; Abraham Duarte and Francisco Parreño; Mauricio Resende, Simone Martins, Luiz Ochi, and Geiza Silva; Xiangjin Lai, Zhipeng Lü, and Jin-Kao Hao; Roberto Aringhieri and Roberto Cordone; Oleg Prokopyev; Jack Brimberg, Nenad Mladenović, Raca Todosijević, and Dragan Urošević. Thank you for your contributions to the area, and in particular to this paper.

This research was funded by "Ministerio de Ciencia, Innovación y Universidades" under grant ref. PGC2018-0953322-B-C21 and PGC2018-0953322-B-C22, "Comunidad de Madrid" and "Fondos Estructurales" of European Union with grant refs. S2018/TCS-4566, Y2018/EMT-5062.

References

Ağca, S., Eksioglu, B., & Ghosh, J. B. (2000). Lagrangian solution of maximum dispersion problems. *Naval Research Logistics, 47*(2), 97–114.
Amirgaliyeva, Z., Mladenović, N., Todosijević, R., & Urošević, D. (2017). Solving the maximum min-sum dispersion by alternating formulations of two different problems. *European Journal of Operational Research, 260*(2), 444–459.
Aringhieri, R., & Cordone, R. (2006). Better and Faster Solutions for the Maximum Diversity Problem. *Technical Report*. Université degli Studi di Milano, Polo Didattico e di Ricerca di Crema Milano.
Aringhieri, R., & Cordone, R. (2011). Comparing local search metaheuristics for the maximum diversity problem. *Journal of the Operational Research Society, 62*(2), 266–280.
Aringhieri, R., Cordone, R., & Grosso, A. (2015). Construction and improvement algorithms for dispersion problems. *European Journal of Operational Research, 242*(1), 21–33.
Aringhieri, R., Cordone, R., & Melzani, Y. (2008). Tabu search versus GRASP for the maximum diversity problem. *4OR, 6*(1), 45–60.
Beasley, J. E. (1990). OR-library: Distributing test problems by electronic mail. *Journal of the Operational Research Society, 41*(11), 1069–1072.
Brimberg, J., Mladenović, N., Todosijević, R., & Urošević, D. (2019). Solving the capacitated clustering problem with variable neighborhood search. *Annals of Operations Research, 272*(1–2), 289–321.
Brimberg, J., Mladenović, N., Urošević, D., & Ngai, E. (2009). Variable neighborhood search for the heaviest -subgraph. *Computers and Operations Research, 36*(11), 2885–2891.
Carrasco, R., Pham, A., Gallego, M., Gortázar, F., Martí, R., & Duarte, A. (2015). Tabu search for the max-mean dispersion problem. *Knowledge-Based Systems, 85*, 256–264.
Chandra, B., & Halldórsson, M. M. (2001). Approximation algorithms for dispersion problems. *Journal of Algorithms, 38*(2), 438–465.
Chandrasekaran, R., & Daughety, A. (1981). Location on tree networks: P-centre and n-dispersion problems. *Mathematics of Operations Research, 6*(1), 50–57.
Church, R. L., & Garfinkel, R. S. (1978). Locating an obnoxious facility on a network. *Transportation Science, 12*(2), 107–118.
Colmenar, J. M., Martí, R., & Duarte, A. (2018). Heuristics for the bi-objective diversity problem. *Expert Systems with Applications, 108*, 193–205.
De Freitas, A., Guimarães, F., Pedrosa Silva, R., & Souza, M. (2014). Memetic self-adaptive evolution strategies applied to the maximum diversity problem. *Optimization Letters, 8*(2), 705–714.
Della Croce, F., Garraffa, M., & Salassa, F. (2016). A hybrid three-phase approach for the max-mean dispersion problem. *Computers and Operations Research, 71*, 16–22.
Dhir, K., Glover, F., & Kuo, C.-C. (1993). Optimizing diversity for engineering management. In *Proceedings of engineering management society conference on managing projects in a borderless world* (pp. 23–26). IEEE.
Duarte, A., & Martí, R. (2007). Tabu search and GRASP for the maximum diversity problem. *European Journal of Operational Research, 178*(1), 71–84.
Duarte, A., Sánchez-Oro, J., Resende, M. G., Glover, F., & Martí, R. (2015). Greedy randomized adaptive search procedure with exterior path relinking for differential dispersion minimization. *Information Sciences, 296*, 46–60.
Erkut, E. (1990). The discrete p-dispersion problem. *European Journal of Operational Research, 46*(1), 48–60.
Erkut, E., & Neuman, S. (1989). Analytical models for locating undesirable facilities. *European Journal of Operational Research, 40*(3), 275–291.
Fekete, S., & Meijer, H. (2004). Maximum dispersion and geometric maximum weight cliques. *Algorithmica*, (38), 501–511.
Feo, T., & Resende, M. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization, 6*(2), 109–133.
Festa, P., & Resende, M. G. (2016). GRASP. In R. Martí, P. Panos, & M. G. Resende (Eds.), *Handbook of heuristics: 1–2* (pp. 465–488). Springer International Publishing.
Gallego, M., Duarte, A., Laguna, M., & Martí, R. (2009). Hybrid heuristics for the maximum diversity problem. *Computational Optimization and Applications, 44*(3), 411–426.
Garraffa, M., Della Croce, F., & Salassa, F. (2017). An exact semidefinite programming approach for the max-mean dispersion problem. *Journal of Combinatorial Optimization, 34*(1), 71–93.
Ghosh, J. B. (1996). Computational aspects of the maximum diversity problem. *Operations Research Letters, 19*(4), 175–181.
Glover, F., Campos, V., & Martí, R. (2021). Tabu search tutorial. A graph drawing application. *TOP*, 1–32.
Glover, F., Kuo, C.-C., & Dhir, K. S. (1995). A discrete optimization model for preserving biological diversity. *Applied Mathematical Modelling, 19*(11), 696–701.
Glover, F., Kuo, C.-C., & Dhir, K. S. (1998). Heuristic algorithms for the maximum diversity problem. *Journal of Information and Optimization Sciences, 19*(1), 109–132.
Glover, F., & Laguna, M. (1998). Tabu search. In *Handbook of combinatorial optimization* (pp. 2093–2229). Springer US.
Goldman, A., & Dearing, P. (1975). Concepts of optimal location for partially noxious facilities. *Bulletin of the Operational Research Society of America, 23*(1), B85.
Hansen, P., & Mladenović, N. (2005). Variable neighborhood search. In *Search methodologies: Introductory tutorials in optimization and decision support techniques* (pp. 211–238). Springer US.
Hart, J. P., & Shogan, A. W. (1987). Semi-greedy heuristics: An empirical study. *Operations Research Letters, 6*(3), 107–114.
Hong, L., & Page, S. E. (2004). Groups of diverse problem solvers can outperform groups of high-ability problem solvers. *Proceedings of the National Academy of Sciences of the United States of America, 101*(46), 16385–16389.
Katayama, K., & Narihisa, H. (2006). An evolutionary approach for the maximum diversity problem. In *Recent advances in memetic algorithms* (pp. 31–47). Berlin/Heidelberg: Springer-Verlag.
Kincaid, R. K. (1992). Good solutions to discrete noxious location problems via metaheuristics. *Annals of Operations Research, 40*(1), 265–281.
Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671–680.
Kochenberger, G., Hao, J.-K., Glover, F., Lewis, M., Lü, Z., Wang, H., et al. (2014). The unconstrained binary quadratic programming problem: A survey. *Journal of combinatorial optimization, 28*(1), 58–81.
Kuby, M. J. (1988). Programming models for facility dispersion: The p-dispersion and maxisum dispersion problems. *Mathematical and Computer Modelling, 10*(10), 792.
Kuo, C.-C., Glover, F., & Dhir, K. S. (1993). Analyzing and modeling the maximum diversity problem by zero-one programming. *Decision Sciences, 24*(6), 1171–1185.
Laguna, M., & Martí, R. (1999). GRASP and path relinking for 2-layer straight line crossing minimization. *INFORMS Journal on Computing, 11*(1), 44–52.
Lai, X., Hao, J., Yue, D., & Gao, H. (2019). Diversification-driven memetic algorithm for the maximum diversity problem. In *Proceedings of 2018 5th IEEE international conference on cloud computing and intelligence systems, CCIS 2018* (pp. 310–314). Institute of Electrical and Electronics Engineers Inc..
Lai, X., & Hao, J.-K. (2016). A tabu search based memetic algorithm for the max-mean dispersion problem. *Computers and Operations Research, 72*, 118–127.
Lai, X., Hao, J.-K., & Glover, F. (2020). A study of two evolutionary/tabu search approaches for the generalized max-mean dispersion problem. *Expert Systems with Applications, 139*, 112856.
Lai, X., Yue, D., Hao, J.-K., & Glover, F. (2018). Solution-based tabu search for the maximum min-sum dispersion problem. *Information Sciences, 441*, 79–94.
Lozano, M., Molina, D., & García-Martínez, C. (2011). Iterated greedy for the maximum diversity problem. *European Journal of Operational Research, 214*(1), 31–38.
Macambira, E. M. (2002). An application of tabu search heuristic for the maximum edge-weighted subgraph problem. *Annals of Operations Research, 117*(1–4), 175–190.
Martí, R., Gallego, M., & Duarte, A. (2010). A branch and bound algorithm for the maximum diversity problem. *European Journal of Operational Research, 200*(1), 36–44.
Martí, R., Gallego, M., Duarte, A., & Pardo, E. G. (2013). Heuristics and metaheuristics for the maximum diversity problem. *Journal of Heuristics, 19*(4), 591–615.
Martí, R., Martínez-Gavara, A., & Sánchez-Oro, J. (2021). The capacitated dispersion problem: An optimization model and a memetic algorithm. *Memetic Computing, 13*, 131–146.
Martí, R., & Sandoya, F. (2013). GRASP and path relinking for the equitable dispersion problem. *Computers and Operations Research, 40*(12), 3091–3099.
Martínez-Gavara, A., Campos, V., Laguna, M., & Martí, R. (2017). Heuristic solution approaches for the maximum minsum dispersion problem. *Journal of Global Optimization, 67*(3), 671–686.
Martínez-Gavara, A., Corberán, T., & Martí, R. (2021). GRASP and tabu search for the generalized dispersion problem. *Expert Systems with Applications, 173*, 114703.
Mladenović, N., Todosijević, R., & Urošević, D. (2016). Less is more: Basic variable neighborhood search for minimum differential dispersion problem. *Information Sciences, 326*, 160–171.
Moon, I. D., & Chaudhry, S. S. (1984). An analysis of network location problems with distance constraints. *Management Science, 30*(3), 290–307.
Palubeckis, G. (2007). Iterated tabu search for the maximum diversity problem. *Applied Mathematics and Computation, 189*(1), 371–383.
Parreño, F., Álvarez-Valdés, R., & Martí, R. (2021). Measuring diversity. A review and an empirical analysis. *European Journal of Operational Research, 289*(2), 515–532.

Pearce, D. (1987). Economics and genetic diversity. *Futures, 19*(6), 710–712.
Peiró, J., Jiménez, I., Laguardia, J., & Martí, R. (2021). Heuristics for the capacitated dispersion problem. *International Transactions in Operational Research, 28*(1), 119–141.
Pisinger, D. (2006). Upper bounds and exact algorithms for p-dispersion problems. *Computers and Operations Research, 33*(5), 1380–1398.
Porter, W., Rawal, K., Rachie, K., Wien, H., & Williams, R. (1975). Cowpea germplasm catalog no 1. In *International institute of tropical agriculture, Ibadan, Nigeria*.
Porumbel, D. C., Hao, J. K., & Glover, F. (2011). A simple and effective algorithm for the MaxMin diversity problem. *Annals of Operations Research, 186*(1), 275–293.
Prokopyev, O. A., Kong, N., & Martinez-Torres, D. L. (2009). The equitable dispersion problem. *European Journal of Operational Research, 197*(1), 59–67.
Resende, M. G., Martí, R., Gallego, M., & Duarte, A. (2010). Grasp and path relinking for the max–min diversity problem. *Computers and Operations Research, 37*(3), 498–508.
Rosenkrantz, D. J., Tayi, G. K., & Ravi, S. S. (2000). Facility dispersion problems under capacity and cost constraints. *Journal of Combinatorial Optimization, 4*(1), 7–33.
Santos, L. F., Ribeiro, M. H., Plastino, A., & Martins, S. L. (2005). A hybrid GRASP with data mining for the maximum diversity problem. In M. J. Blesa, C. Blum, A. Roli, & M. Sampels (Eds.), *Lecture notes in computer science (including subseries lecture notes in hybrid metaheuristics)* (pp. 116–127).
Sayah, D., & Irnich, S. (2017). A new compact formulation for the discrete p-dispersion problem. *European Journal of Operational Research, 256*(1), 62–67.
Sayyady, F., & Fathi, Y. (2016). An integer programming approach for solving the p-dispersion problem. *European Journal of Operational Research, 253*(1), 216–225.
Shier, D. R. (1977). A min-max theorem for p-center problems on a tree. *Transportation Science, 11*(3), 243–252.
Silva, G. C., De Andrade, M. R., Ochi, L. S., Martins, S. L., & Plastino, A. (2007). New heuristics for the maximum diversity problem. *Journal of Heuristics, 13*(4), 315–336.
Silva, G. C., Ochi, L. S., & Martins, S. L. (2004). Experimental comparison of greedy randomized adaptive search procedures for the maximum diversity problem. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics): 3059* (pp. 498–512).
Swierenga, R. P. (1977). Ethnicity in historical perspective. *Social Science, 52*(1), 31–44.
Wang, J., Zhou, Y., Cai, Y., & Yin, J. (2012). Learnable tabu search guided by estimation of distribution for maximum diversity problems. *Soft Computing, 16*(4), 711–728.
Wang, Y., Hao, J.-K., Glover, F., & Lü, Z. (2014). A tabu search based memetic algorithm for the maximum diversity problem. *Engineering Applications of Artificial Intelligence, 27*, 103–114.
Wu, Q., & Hao, J.-K. (2013). A hybrid metaheuristic method for the maximum diversity problem. *European Journal of Operational Research, 231*(2), 452–464.
Zhou, Y., & Hao, J. K. (2017). An iterated local search algorithm for the minimum differential dispersion problem. *Knowledge-Based Systems, 125*, 26–38.
Zhou, Y., Hao, J. K., & Duval, B. (2017). Opposition-based memetic search for the maximum diversity problem. *IEEE Transactions on Evolutionary Computation, 21*(5), 731–745.