European Journal of Operational Research 197 (2009) 59–67

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

The equitable dispersion problem

Oleg A. Prokopyev [a,*], Nan Kong [b], Dayna L. Martinez-Torres [c]

[a] Department of Industrial Engineering, University of Pittsburgh, Pittsburgh, PA 15261, United States
[b] Weldon School of Biomedical Engineering, Purdue University, West Lafayette, IN 47907, United States
[c] Department of Industrial and Management Systems Engineering, University of South Florida, Tampa, FL 33620, United States

ARTICLE INFO

Article history:
Received 20 July 2007
Accepted 4 June 2008
Available online 13 June 2008

Keywords:
Equitable dispersion
Maximum dispersion
Equity
Binary nonlinear programming

ABSTRACT

Most optimization problems focus on efficiency-based objectives. Given the increasing awareness of system inequity resulting from solely pursuing efficiency, we conceptualize a number of new element-based equity-oriented measures in the dispersion context. We propose the equitable dispersion problem that maximizes the equity among elements based on the introduced measures in a system defined by inter-element distances. Given the proposed optimization framework, we develop corresponding mathematical programming formulations as well as their mixed-integer linear reformulations. We also discuss computational complexity issues, related graph-theoretic interpretations and provide some preliminary computational results.

1. Introduction

Suppose, we are given a set N of n elements, and d_{ij}, the *inter-element distance* between any two elements i and j. The so-called *Maximum Dispersion Problem (Max DP)* is to select a subset $M \subseteq N$ of cardinality m, i.e., $|M| = m$, such that some efficiency-based function $z(M)$ of d_{ij} from the selected subset M is maximized. The problem is then formulated as $\max_{M \subseteq N}\{z(M) : M \in \mathscr{F}\}$, where $\mathscr{F}$ is the set containing all subsets of N. Two important and widely addressed functions in the literature (see, e.g., [3]) are the *sum* and *minimum* of d_{ij} for all $i,j \in M$, respectively. The former and latter functions can thus be written as $\sum_{i<j;i,j\in M} d_{ij}$ and $\min_{i<j;i,j\in M} d_{ij}$, where it is usually assumed that for all $i,j \in N$, we have $d_{ij} = d_{ji}$. Hence, the first selection criterion for the maximally disperse set is called *Maxsum* and the second criterion is called *Maxmin*. To differentiate the two resultant maximization problems, we name them accordingly as *Maxsum DP* and *Maxmin DP*.

Define a binary variable $x_i = 1$ if element i belongs to M and $x_i = 0$ otherwise, for $i = 1, \ldots, n$. Then the problem with the former objective can be rewritten as a cardinality-constrained quadratic 0–1 programming problem:

$$\max_{x\in\{0,1\}^n} \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij}x_i x_j \tag{1}$$

$$\text{s.t.} \quad \textstyle\sum_{i=1}^{n} x_i = m; \tag{2}$$

$$x_i \in \{0,1\}, \quad i = 1,\ldots,n. \tag{3}$$

Problems (1)–(3) sometimes is also referred to as *maximum diversity/similarity problem* [15], where d_{ij} measures the difference/similarity between elements i and j. Problems of type (1)–(3) can be tackled using algorithms designed for solving quadratic 0–1 programs [8,25], or we can easily obtain a classical linear mixed 0–1 reformulation (see e.g., [17,23]), which, in turn, can be solved using any standard mixed integer programming (MIP) solver (e.g., CPLEX, XPRESS-MP):

$$\max_{x\in\{0,1\}^n} \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij}z_{ij}$$

$$\text{s.t.} \quad (2), (3), \text{ and}$$

$$z_{ij} \geqslant x_i + x_j - 1, \quad z_{ij} \leqslant x_i, \quad z_{ij} \leqslant x_j, \quad 1 \leqslant i < j \leqslant n; \tag{4}$$

$$z_{ij} \in \{0,1\}, \quad 1 \leqslant i < j \leqslant n. \tag{5}$$

In the literature, the maximum dispersion problem with the second objective is also referred to as the *p-dispersion problem* [12,22,24]. It is regarded as a maxmin problem among individual inter-element distances. A mixed-integer 0–1 reformulation of the problem is presented as:

$$\max \quad y, \tag{6}$$

$$\text{s.t.} \quad (2)-(5), \text{ and}$$

$$y \leqslant d_{ij} + (1 - z_{ij})b_{ij}, \quad 1 \leqslant i < j \leqslant n. \tag{7}$$

Note that b_{ij} is conveniently chosen as $d^U - d_{ij}$, where d^U is an upper bound on the optimal value of y, e.g., $d^U = \max_{i,j\in N} d_{ij}$. The above linearization is straightforward and thus the formulations may be tightened. For detailed development of the above formulations and further tightening, we refer the readers to [3].

* Corresponding author. Tel.: +1 412 624 9833.
E-mail address: prokopyev@engr.pitt.edu (O.A. Prokopyev).

0377-2217/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2008.06.005

As discussed in [15,23], an element i may be characterized by a vector with ℓ possible attributes $(a_{i1}, \ldots, a_{i\ell})$. Then for d_{ij} we can consider, for instance, a Euclidean distance measure, or a general L_p norm:

$$d_{ij} = \left(\sum_{s=1}^{\ell} |a_{is} - a_{js}|^p \right)^{1/p}. \tag{8}$$

In the case of (8), all d_{ij}'s are nonnegative. However, d_{ij} can generally take both positive and negative values.

The maximum dispersion problem primarily focuses on operational efficiency of locating facilities according to distance, accessibility, impacts, etc. (see [11,12,22,29]). It also arises in various other contexts including maximally diverse/similar group selection (e.g., biological diversity, admissions policy formulation, committee formation, curriculum design, market planning, etc.) [1,15,16,23,34] and densest subgraph identification [21]. For a detailed literature survey on *Maxsum DP* and *Maxmin DP*, we refer the readers to [3,11,23]. For these two problems, exact methods [3,15,28] and heuristics [15,18,20,30,31] have been developed.

An equally important concern in facility location theory is *equity*. It is here referred to as the concept or idea of fairness among candidate facility sites. Furthermore, one particular interest is in balancing equity with efficiency in a model-building paradigm [33]. However, the majority of models in the operations research literature are efficiency-based and little work has been done for the equity concern [26]. This is especially critical when dealing with urban public facility location [33]. This paper explores the use of several equity measures in the framework of the dispersion problem. These equity measures may also arise in the contexts of diverse/similar group selection, and dense and regular subgraph identification. For example, one may address fair diversification or assimilation among members of a network. Somewhat related work on equity based measures in network flow problems can be found in [6,7].

The remainder of this paper is organized as follows. Section 2 introduces a number of new measures in the context of the dispersion problem. Section 3 develops a nonlinear binary optimization framework with the introduced equity measures being objectives. Section 4 presents the corresponding linear mixed 0–1 reformulations. Section 5 shows the computational complexity of the resultant optimization problems. Section 6 provides several graph-theoretic results related to the measures. Section 7 discusses our preliminary computational experiments. Section 8 concludes the paper and points out future research directions.

2. Equity measures

As discussed in Section 1, the dispersion problem in the literature has mainly focused on efficiency-based objectives. In this section, we introduce several equity-based measures, each of which alternatively represents some function of dispersion with respect to individual elements. Therefore, unlike efficiency-based measures that consider some dispersion quantity for the entire selection M (e.g., total amount of dispersion and the minimum level of dispersion), these measures are used to balance the dispersion among individual elements in M. To ensure equity among elements, one may input some of these measures into an optimization framework. Solving the resultant optimization problem would, in some sense, achieve equitable dispersion among the selected elements.

- *Mean Dispersion Measure*. For each $M \subseteq N$, let us define

  $$f_1(M) = \left(\sum_{i<j;i,j\in M} d_{ij} \right) \Big/ |M| = z(M)/|M|.$$

  Generically speaking, $f_1(M)$ is an average efficiency measure. In this context, we call it *Mean Dispersion Measure*. One way to ensure equity is to maximize $f_1(M)$.
- *Generalized Mean Dispersion Measure*. For each $M \subseteq N$, let us define

  $$f_2(M) = \left(\sum_{i<j;i,j\in M} d_{ij} \right) \Big/ \left(\sum_{i\in M} w_i \right).$$

  The value w_i can be considered as the weight assigned to element i, $i \in N$. Hence, this measure is the average weighted efficiency measure. One may also ensure equity by maximizing $f_2(M)$.
- *Minsum Dispersion Measure*. For each $M \subseteq N$, let us define

  $$f_3(M) = \min_{i\in M} \sum_{j\in M, j\neq i} d_{ij}.$$

  This measure can be interpreted as follows. For each element $i \in M$, we can measure the total dispersion associated with i, denoted by $c(M, i)$. That is, $c(M, i) = \sum_{j\in M, j\neq i} d_{ij}$ for all $i \in M$. Then $f_3(M)$ is the smallest value among all selected elements, i.e., $f_3(M) = \min_{i\in M} c(M, i)$. An alternative way to ensure equity is to maximize $f_3(M)$, which forms a maxmin framework. Therefore, we term this measure *Minsum Dispersion Measure*. Note that this measure considers the minimum aggregate dispersion among elements, which is different from the one presented in (6) and (7) (the p-dispersion problem in [12]).
- *Differential Dispersion Measure*. For each $M \subseteq N$, let us define

  $$f_4(M) = \max_{i\in M} \sum_{j\in M, j\neq i} d_{ij} - \min_{i\in M} \sum_{j\in M, j\neq i} d_{ij}.$$

  This measure can be understood as the difference between the largest and smallest $c(M, i)$ values, i.e., $f_4(M) = \max_{i\in M} c(M, i) - \min_{i\in M} c(M, i)$. Therefore, we term this measure *Differential Dispersion Measure*. A third way to ensure equity is to minimize $f_4(M)$.

To summarize, the first equity measure introduced in this section addresses the average efficiency level whereas the last two consider the element(s) with extreme efficiency level(s). It is clear that these measures are based on different interpretations of equity.

3. Mathematical programming models

After defining the equity measures in the previous section, we can next formulate respective mathematical programs. Except for the first mathematical program, we impose the cardinality restriction on the number of elements to be selected. Note that in the first model, the group size is part of the decision to make.

3.1. Maximum mean dispersion problem (Max-Mean DP)

This problem can be formulated as the following fractional 0–1 programming problem:

$$\begin{aligned} \max_{x\in\{0,1\}^n} \quad & \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_{ij} x_i x_j}{\sum_{i=1}^{n} x_i} \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i \geqslant 1. \end{aligned} \tag{9}$$

As stated earlier, we do not impose the cardinality restriction $\sum_{i=1}^{n} x_i = m$ in the above formulation. It is clear that the Max-Mean DP is reduced to Maxsum DP once the number of elements to be selected is fixed.

The generalized version of the Max-Mean DP, termed the *Generalized Max-Mean DP*, can be stated as:

$$\begin{aligned} \max_{x\in\{0,1\}^n} \quad & \frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij}x_ix_j}{\sum_{i=1}^{n} w_ix_i} \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i \geqslant 1. \end{aligned} \tag{10}$$

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

3.2. Maximum minsum dispersion problem (Max-Minsum DP)

The mathematical programming problem corresponding to the *minsum dispersion* measure is formulated as:

$$\begin{aligned} \max_{x\in\{0,1\}^n} \quad & \left\{ \min_{i:x_i=1} \sum_{j:j\neq i} d_{ij}x_j \right\} \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i = m. \end{aligned} \tag{11}$$

Problem (11) can be equivalently rewritten as

$$\begin{aligned} \max_{x\in\{0,1\}^n} \quad & \left\{ \min_{i} \left\{ \sum_{j:j\neq i} d_{ij}x_jx_i + M^+(1-x_i) \right\} \right\} \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i = m \end{aligned} \tag{12}$$

where M^+ is a large enough constant, e.g.,

$$M^+ = 1 + \max_i \sum_{j:j\neq i} \max\{d_{ij}, 0\}. \tag{13}$$

3.3. Minimum differential dispersion problem (Min-Diff DP)

The problem of finding the best subset $M \subseteq N$ with respect to this measure can be formulated as the following 0–1 programming problem:

$$\begin{aligned} \min_{x\in\{0,1\}^n} \quad & \left\{ \max_{i:x_i=1} \sum_{j:j\neq i} d_{ij}x_j - \min_{i:x_i=1} \sum_{j:j\neq i} d_{ij}x_j \right\} \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i = m. \end{aligned} \tag{14}$$

An equivalent big-M formulation is given by

$$\begin{aligned} \min_{x} \quad & \left\{ \max_{i} \left\{ \sum_{j:j\neq i} d_{ij}x_jx_i + M^-(1-x_i) \right\} \right. \\ & \left. - \min_{i} \left\{ \sum_{j:j\neq i} d_{ij}x_jx_i + M^+(1-x_i) \right\} \right\} \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i = m, x \in \{0,1\}^n, \end{aligned} \tag{15}$$

where M^+ can be selected according to (13) and

$$M^- = -1 + \min_i \sum_{j:j\neq i} \min\{d_{ij}, 0\}. \tag{16}$$

The mathematical programs developed in this section are nonlinear binary programs. We linearize these models in the next section by introducing auxiliary variables.

4. Basic linear mixed 0–1 reformulations

The development of exact linear reformulations facilitates the use of state-of-the-art mixed-integer linear programming solvers. In order to linearize the problems introduced in Section 3, we can apply techniques similar to those used in [3,23,32,35]. Applying more specialized techniques may derive tighter mixed-integer programming formulations. Since the focus of this paper is to establish the equity measures in the context of the dispersion problem, we leave formulation tightening to future research. Once a respective linear mixed-integer 0–1 reformulation is developed, we also provide the equivalence proof between the reformulation and its nonlinear counterpart.

4.1. Max-Mean DP

Proposition 1 ([32,35]). *A polynomial mixed 0–1 term $z = xy$, where x is a 0–1 variable, and y is a nonnegative continuous variable, can be equivalently represented by the following four linear inequalities: (1) $z \geqslant y - U(1-x)$; (2) $z \leqslant y$; (3) $z \leqslant Ux$; (4) $z \geqslant 0$, where U is an upper bound on y, i.e., $0 \leqslant y \leqslant U$.*

The next proposition immediately follows from Proposition 1.

Proposition 2. *A polynomial mixed 0–1 term $z = x_1x_2y$, where x_1 and x_2 are 0–1 variables, and y is a nonnegative continuous variable, can be equivalently represented by the following five linear inequalities: (1) $z \geqslant y - U(2 - x_1 - x_2)$; (2) $z \leqslant y$; (3) $z \leqslant Ux_1$; (4) $z \leqslant Ux_2$; (5) $z \geqslant 0$, where U is an upper bound on y, i.e., $0 \leqslant y \leqslant U$.*

In order to reformulate (9) as a linear mixed 0–1 programming problem, let us define a new variable y such that

$$y = \frac{1}{\sum_{i=1}^{n} x_i}. \tag{17}$$

This definition is equivalent to

$$\sum_{i=1}^{n} x_iy = 1. \tag{18}$$

With the newly introduced variable y, problem (9) can be rewritten as

$$\max_{x,y} \quad \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij}x_ix_jy \tag{19}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} x_i \geqslant 1, \quad \sum_{i=1}^{n} x_iy = 1, \quad x \in \{0,1\}^n. \tag{20}$$

Next, nonlinear terms x_iy in (20) and x_ix_jy in (19) can be linearized by introducing additional variables z_i and z_{ij}, and applying the results of Propositions 1 and 2. The total number of new variables z_i, z_{ij}, and y, is $n(n+1)/2 + 1$. The parameter U can be set to 1.

Therefore, the linear mixed 0–1 programming reformulation is presented as:

$$\max_{x,z,y} \quad \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij}z_{ij} \tag{21}$$

$$\text{s.t.} \quad y - z_i \leqslant 1 - x_i, \quad z_i \leqslant y, \quad z_i \leqslant x_i, \quad z_i \geqslant 0, \quad i = 1,\ldots,n; \tag{22}$$

$$\begin{aligned} & y - z_{ij} \leqslant 2 - x_i - x_j, \quad z_{ij} \leqslant y, \quad z_{ij} \leqslant x_i, \\ & z_{ij} \leqslant x_j, \quad z_{ij} \geqslant 0, \quad 1 \leqslant i < j \leqslant n; \end{aligned} \tag{23}$$

$$\sum_{i=1}^{n} x_i \geqslant 1; \quad \sum_{i=1}^{n} z_i = 1; \quad x \in \{0,1\}^n. \tag{24}$$

Proposition 3. *Formulations (9) and (21)–(24) are equivalent, i.e., (9) $\Leftrightarrow$ (21)–(24).*

Proof. The proposition is valid by the construction of (21)–(24). □

Obviously, if $d_{ij} \geqslant 0$ for all $i,j = 1,\ldots,n$, (23) can be simplified and replaced by

$$z_{ij} \leqslant y, \quad z_{ij} \leqslant x_i, \quad z_{ij} \leqslant x_j, \quad 1 \leqslant i < j \leqslant n. \tag{25}$$

The linear mixed 0–1 formulation for (10) can be easily obtained in a similar manner.

4.2. Max-Minsum DP

Let L_i and U_i be lower and upper bounds on the value of $\sum_{j:j\neq i} d_{ij}x_j$, which can be set, for example, as

$$L_i = \sum_{j:j\neq i} \min\{d_{ij}, 0\}, \quad U_i = \sum_{j:j\neq i} \max\{d_{ij}, 0\}. \tag{26}$$

After some simplification problem (12) can be further written as:

$$\max_{r,x} \quad r \tag{27}$$

$$\text{s.t.} \quad r \leqslant \sum_{j:j\neq i} d_{ij}x_j - L_i(1-x_i) + M^+(1-x_i), \quad i = 1,\ldots,n; \tag{28}$$

$$\sum_{i=1}^{n} x_i = m; \quad x \in \{0,1\}^n, \tag{29}$$

where M^+ is a sufficiently large constant defined earlier in (13).

Proposition 4. *Formulations (11), (12) and (27)–(29) are equivalent, i.e., (11) $\Leftrightarrow$ (12) $\Leftrightarrow$ (27)–(29).*

Proof

- (11) $\Leftrightarrow$ (12): By the selection of M^+ in (13), it follows that if

$$i^* = \arg\min_i \left\{ \sum_{j:j\neq i} d_{ij}x_jx_i + M^+(1-x_i) \right\},$$

  then $x_{i^*} = 1$, which immediately implies (11) $\Leftarrow$ (12). The proof of the opposite direction (11) $\Rightarrow$ (12) clearly follows from the selection of M^+.
- (11) $\Leftrightarrow$ (27)–(29): First, let us prove (11) $\Leftarrow$ (27)–(29). Let r^* and x^* be the solution of 27. Consider all i's such that $x_i^* = 1$. Constraints (28) become

$$r^* \leqslant \sum_{j:j\neq i} d_{ij}x_j^* \quad \forall i \in \{i = 1,\ldots,n: \ x_i^* = 1\}. \tag{30}$$

Then from inequalities (30), it follows that

$$r^* \leqslant \min_{i:x_i^*=1} \sum_{j:j\neq i} d_{ij}x_j^*. \tag{31}$$

Next consider all i's such that $x_i^* = 0$. Constraints (28) become

$$r^* \leqslant \sum_{j:j\neq i} d_{ij}x_j^* - L_i + M^+ \quad \forall i \in \{i = 1,\ldots,n: \ x_i^* = 0\}. \tag{32}$$

By the definition of L_i, we have $\sum_{j:j\neq i} d_{ij}x_j^* \geqslant L_i$. Therefore, by the definition of M^+ constraints (32) can be dropped since they are weaker than (31). Since we maximize r in the objective function (27) then from (31) we conclude

$$r^* = \min_{i:x_i^*=1} \sum_{j:j\neq i} d_{ij}x_j^*. \tag{33}$$

The necessary result follows from (27), (33), and the definitions of r^* and x^*.The proof of (11) $\Rightarrow$ (27)–(29) follows from the construction of (27)–(29). □

4.3. Min-Diff DP

A linear mixed 0–1 reformulation of (14) is given by

$$\min_{t,r,s,x} \quad t \tag{34}$$

$$\text{s.t.} \quad t \geqslant r - s, \quad i = 1,\ldots,n; \tag{35}$$

$$r \geqslant \sum_{j:j\neq i} d_{ij}x_j - U_i(1-x_i) + M^-(1-x_i), \quad i = 1,\ldots,n; \tag{36}$$

$$s \leqslant \sum_{j:j\neq i} d_{ij}x_j - L_i(1-x_i) + M^+(1-x_i), \quad i = 1,\ldots,n; \tag{37}$$

$$\sum_{i=1}^{n} x_i = m; \quad x \in \{0,1\}^n. \tag{38}$$

Proposition 5. *Formulations (14), (15), and (34)–(38) are equivalent, i.e., (14) $\Leftrightarrow$ (15) $\Leftrightarrow$ (34)–(38).*

Proof.

- (14) $\Leftrightarrow$ (15): By the selection of M^+ and M^- (see (13) and (16)), it follows that
  – if $i^* = \arg\min_i\{\sum_{j:j\neq i} d_{ij}x_jx_i + M^+(1-x_i)\}$, then $x_{i^*} = 1$, and
  – if $i^* = \arg\max_i\{\sum_{j:j\neq i} d_{ij}x_jx_i + M^-(1-x_i)\}$, then $x_{i^*} = 1$,which immediately implies (14) $\Leftarrow$ (15). The proof of the opposite direction (14) $\Rightarrow$ (15) clearly follows from the selection of M^+ and M^-.
- (14) $\Leftrightarrow$ (34)–(38): We first prove (14) $\Leftarrow$ (34)–(38). Let t^*, r^*, s^*, and x^* be the solution of 34. For all i's such that $x_i^* = 1$, constraints (36) and (37) become

$$r^* \geqslant \sum_{j:j\neq i} d_{ij}x_j^* \quad \forall i \in \{i = 1,\ldots,n: \ x_i^* = 1\}, \tag{39}$$

$$s^* \leqslant \sum_{j:j\neq i} d_{ij}x_j^* \quad \forall i \in \{i = 1,\ldots,n: \ x_i^* = 1\}. \tag{40}$$

By the definitions of L_i and U_i, inequalities (39) and (40) imply

$$r^* \geqslant \max_{i:x_i^*=1} \sum_{j:j\neq i} d_{ij}x_j^* \quad \text{and} \quad s^* \leqslant \min_{i:x_i^*=1} \sum_{j:j\neq i} d_{ij}x_j^*. \tag{41}$$

Similarly, for all i's such that $x_i^* = 0$, constraints (36) and (37) become

$$r^* \geqslant \sum_{j:j\neq i} d_{ij}x_j^* - U_i + M^- \quad \forall i \in \{i = 1,\ldots,n: \ x_i^* = 0\}, \tag{42}$$

$$s^* \leqslant \sum_{j:j\neq i} d_{ij}x_j^* - L_i + M^+ \quad \forall i \in \{i = 1,\ldots,n: \ x_i^* = 0\}. \tag{43}$$

By the definitions of L_i and U_i, we have $\sum_{j:j\neq i} d_{ij}x_j^* \geqslant L_i$, and $\sum_{j:j\neq i} d_{ij}x_j^* \leqslant U_i$, respectively. Therefore, constraints (42) and (43) can be dropped since they are weaker than (41). Constraint (35) enforces that the minimization of t in the objective function (34) will correspond to the minimum value of r and maximum value of s. Therefore, from (41) we conclude that

$$t^* = r^* - s^*, \ r^* = \max_{i:x_i^*=1} \sum_{j:j\neq i} d_{ij}x_j^* \text{ and } s^* = \min_{i:x_i^*=1} \sum_{j:j\neq i} d_{ij}x_j^*. \tag{44}$$

The necessary result follows from (34), (44), and the definitions of t^*, r^*, s^*, and x^*.The proof of (14) $\Rightarrow$ (34)–(38) follows from the construction of (34)–(38). □

5. Computational complexity

It is known that both *Maxsum DP* and *Maxmin DP* are strongly *NP*-hard [11,15,23]. In this section we study the computational complexity of *Max-Mean DP* (problem (9)), *Max-Minsum DP* (problem (11)), and *Min-Diff DP* (problem (14)).

If $d_{ij} \geqslant 0$, *Max-Mean DP* is polynomially solvable (see [27]). We next show that the problem becomes difficult when no constraints are imposed on the sign of d_{ij}.

Proposition 6. *Max-Mean DP is strongly NP-hard if coefficients d_{ij} can take both positive and negative values.*

Proof. The proof of the result is based on the *MAXIMUM CLIQUE* problem known to be *NP*-complete [14]. Let $G = (V, E)$ be an undirected graph with n nodes ($|V| = n$). A subset of nodes $C \subseteq V$ is called a *clique* of the graph if for any two nodes v_1 and v_2 that belong to C, i.e., $v_1, v_2 \in C \subseteq V$, there is an edge $(v_1, v_2) \in E$ connecting them. The decision version of *MAXIMUM CLIQUE* consists of checking whether there exists a clique C of size at least K in G. For an extensive survey on the maximum clique problem, we refer the readers to [5].

Consider the following instance of *Max-Mean DP*:

$$\max_{x\in\{0,1\}^n, x\neq 0} g(x) = \frac{-n^2\sum_{(i,j)\notin E, i<j} x_i x_j + \sum_{(i,j)\in E, i<j} x_i x_j}{\sum_{i=1}^n x_i}, \tag{45}$$

where $d_{ij} = 1$ for $(i,j) \in E$ and $d_{ij} = -n^2$ for $(i,j) \notin E$. Let x^* be a nonzero 0–1 vector. Define a subgraph $C(V_c, E_c) \subseteq G$ induced by nodes $V_c = \{i \in \{1,\dots,n\} : x_i^* = 1\}$. Notice that if $x_i^* = 1$, $x_j^* = 1$ and $(i,j) \notin E$ (i.e., C is not a clique) then $g(x^*) < 0$. If C is a clique, i.e., for any $x_i^* = 1$ and $x_j^* = 1$ we have $(i,j) \in E$, then

$$g(x^*) = \frac{|C|\cdot(|C|-1)}{2|C|} = \frac{|C|-1}{2} \geqslant 0.$$

Therefore, graph G contains a clique of size at least K if and only if

$$\max_{x\in\{0,1\}^n,\ x\neq 0} g(x) \geqslant \frac{K-1}{2}. \qquad \square$$

Next we show that *Max-Minsum DP* and *Min-Diff DP* are strongly *NP*-hard as well.

Proposition 7. *Max-Minsum DP is strongly NP-hard.*

Proof. As in the previous proof, we use a reduction from the *MAXIMUM CLIQUE* problem. Given an undirected graph $G(V,E)$, let $d_{ij} = 1$ if $(i,j) \in E$ and 0 otherwise. For a given nonzero vector x^* such that $\sum_i x_i^* = K$, define a subgraph $C(V_c, E_c) \subseteq G$ induced by nodes $V_c = \{i \in \{1,\dots,n\} : x_i^* = 1\}$, and thus $|V_c| = K$. Observe that C is a clique if and only if for all $i \in V_c$, we have $\sum_{j:j\neq i} d_{ij} x_j^* = K-1$. Therefore, $G(V,E)$ contains a clique of size at least K if and only if the solution of the *Max-Minsum DP* instance, with $m = K$ and earlier specified d_{ij}'s, satisfies

$$\max_{x\in\{0,1\}^n} \min_{i:x_i=1} \sum_{j:j\neq i} d_{ij} x_j \geqslant K-1. \qquad \square$$

Proposition 8. *Min-Diff DP is strongly NP-hard.*

Proof. We use the reduction from the *MAXIMUM INDEPENDENT SET* problem, which is *NP*-complete [14]. For an undirected graph $G = (V,E)$, a subset of nodes $I \subseteq V$ is called an *independent set* of the graph G if for any two nodes v_1 and v_2 from I, i.e., $v_1, v_2 \in I \subseteq V$, $(v_1, v_2) \notin E$. The decision version of the *MAXIMUM INDEPENDENT SET* problem consists of checking whether there exists an independent set I of size at least K in G.

Let $i,j = 1,\dots,2n$. Define d_{ij} as

- $d_{ij} = 1$ if $1 \leqslant i,j \leqslant n$ and $(i,j) \in E$;
- $d_{ij} = 0$, otherwise.

For a given graph $G(V,E)$ with n nodes ($|V| = n$), we construct an instance of *Min-Diff DP* with $m = n + K$ as:

$$\begin{aligned} &\min_{x\in\{0,1\}^{2n}} \left\{ \max_{i:x_i=1} \sum_{j:j\neq i} d_{ij} x_j - \min_{i:x_i=1} \sum_{j:j\neq i} d_{ij} x_j \right\} \\ &\text{s.t.} \quad \sum_{i=1}^{2n} x_i = n + K. \end{aligned} \tag{46}$$

Let x^* be an optimal solution of (46). Note that since $\sum_{i=1}^{2n} x_i^* = n + K$, there exists an $i > n$ such that $x_i^* = 1$. Therefore

$$\min_{i:x_i^*=1} \sum_{j:j\neq i} d_{ij} x_j^* = 0. \tag{47}$$

For a given x^*, define a subgraph $G(V_I, E_I) \subseteq G$ induced by nodes $V_I = \{i \in \{1,\dots,n\} : x_i^* = 1\}$. Obviously, $|V_I| \geqslant K$. Then V_I is an independent set if and only if

$$\max_{i:\ x_i^*=1} \sum_{j:j\neq i} d_{ij} x_j^* = 0. \tag{48}$$

From (47) and (48) it follows that G contains an independent set of size at least K if and only if the optimal solution of (46) is 0. $\square$

It is interesting that *Max-Minsum DP* and *Min-Diff DP* remain *NP*-hard even if we impose sign restrictions on d_{ij}.

Proposition 9. *Max-Minsum DP and Min-Diff DP remain NP-hard if we impose sign restrictions on d_{ij}, e.g., $\forall i,j,\ d_{ij} \geqslant 0$, or $\forall i,j,\ d_{ij} \leqslant 0$.*

Proof. It is enough to observe that due to a knapsack constraint we can always reduce a general problem of type (1)–(3) to an equivalent problem with sign restrictions on d_{ij} (see, e.g., [19]). Let $\tilde{d}_{ij} = d_{ij} + M$, where $M = \max_{i,j}|d_{ij}|$ if we need $\forall i,j,\ \tilde{d}_{ij} \geqslant 0$ and $M = -\max_{i,j}|d_{ij}|$ if we need $\forall i,j,\ \tilde{d}_{ij} \leqslant 0$. Then

$$\begin{aligned} \sum_{i=1}^n \sum_{j\neq i} d_{ij} x_i x_j &= \sum_{i=1}^n \sum_{j\neq i} (\tilde{d}_{ij} - M) x_i x_j \\ &= \sum_{i=1}^n \sum_{j\neq i} \tilde{d}_{ij} x_i x_j - m(m-1)M. \end{aligned} \tag{49}$$

The term $m(m-1)M$ is a constant. Therefore, we can optimize $\sum\sum \tilde{d}_{ij} x_i x_j$ instead of $\sum\sum d_{ij} x_i x_j$, thus reducing our initial general problem to a more specific case with sign restrictions on the values of d_{ij}. The result follows directly from formulations (12) and (15), which are equivalent to (11) and (14), respectively. $\square$

6. Graph-theoretic interpretation

In this section we will provide interesting interpretations of *Maxsum DP*, *Maxmin DP*, *Max-Mean DP*, *Max-Minsum DP*, and *Min-Diff DP* (see (1)–(3), (6), (7), (9), (11) and (14)) in the graph-theoretic context. For graph-theoretic terms that appear in this section, we refer the readers to detailed discussions in [2,4,10].

Consider an undirected graph $G = (V,E)$, where V denotes the set of nodes and E denotes the set of edges. For each of the problems, we consider two cases as follows:

(a) graph G is unweighted and the matrix $\{d_{ij}\}$ is the adjacency matrix associated with G, i.e., $d_{ij} = 1$ if $(i,j) \in E$ and $d_{ij} = 0$, otherwise; and
(b) graph G is weighted and nonzero d_{ij} defines the edge weight of $(i,j) \in E$, while $d_{ij} = 0$ implies that $(i,j) \notin E$.

In all of problems (1)–(3), (6), (7), (9), (11) and (14), we are looking for a subgraph $\widetilde{G}^* \subseteq G$ induced by a subset of nodes $V(\widetilde{G}^*) = \widetilde{V} \subseteq V$, where node $i \in \widetilde{V}$ if and only if $x_i = 1$ in the solution of the respective problem. Note that the number of nodes in $\widetilde{G}^*$ is m, i.e., $|\widetilde{V}| = m$.

- *Maxsum DP*
  - Unweighted case: the solution of problem (1)–(3) provides a *densest* (i.e., with the largest number of edges) subgraph with exactly m nodes.
  - Weighted case: the solution of problem (1)–(3) provides a *heaviest* (i.e., with the largest sum of weights) subgraph with exactly m nodes.
- *Maxmin DP*
  - Unweighted case: each objective function coefficient of problem (6) and (7) can take only two values: either 0 or 1. The optimal value of (6) is 1 if and only if graph G contains a clique (complete subgraph) of size at least m.
  - Weighted case: we search for a subgraph of size m that maximizes the weight of the least heavy edge of the subgraph.
- *Max-Mean DP*
  - Unweighted case: the *density* $d(G)$ of graph G can be defined as the maximum ratio of the number of edges $e_{\widetilde{G}}$ to the number of nodes $n_{\widetilde{G}}$ over all subgraphs $\widetilde{G} \subseteq G$ (see [27]), i.e.,

$$d(G) = \max_{\widetilde{G} \subseteq G} \frac{e_{\widetilde{G}}}{n_{\widetilde{G}}}. \quad (50)$$

The ratio $e_{\widetilde{G}}/n_{\widetilde{G}}$ is called the *average degree* of $\widetilde{G}$ (see, e.g., [10]). Then an optimal solution of (50) defines a subgraph $\widetilde{G}^* \subseteq G$ with the largest average degree. Problem (50) is equivalent to (9) and thus the solution of (9) identifies a densest subgraph $\widetilde{G}^*$. A detailed description of (50) as well as a polynomial time algorithm for its solution (the algorithm requires at most $O(n^4)$ operations) are discussed in [27].

– Weighted case: if we assign an arbitrary nonnegative value to d_{ij}, as the weight of edge (i,j), then the solution of problem (9) yields the *weighted density* of graph G.

- *Max-Minsum DP*
  – Unweighted case: it is easy to notice that if $x_i = 1$ then $\sum_{j \neq i} d_{ij} x_j$ defines the *degree* of a node i in the selected subgraph. Usually the minimum degree of a node in graph G is denoted by $\delta(G)$ (see, e.g., [4]). Therefore, in (11) we search for a subgraph $\widetilde{G}^* \subseteq G$ with exactly m nodes that maximizes $\delta(\cdot)$, i.e.,

$$\begin{aligned} \max_{\widetilde{G} \subseteq G} \quad & \delta(\widetilde{G}) \\ \text{s.t.} \quad & |V(\widetilde{G})| = m. \end{aligned} \quad (51)$$

We can also observe that if the solution of (51) is $m-1$ then $\widetilde{G}^*$ is a *clique*.

  – Weighted case: if $x_i = 1$ then $\sum_{j \neq i} d_{ij} x_j$ defines the total weight of edges incident to node i in the selected subgraph. This set of edges is usually referred to as the *incidence set* of i in the subgraph. Therefore, the solution of (11) identifies a subgraph $\widetilde{G}^* \subseteq G$ with m nodes that maximizes the minimum total weight among incidence sets in $\widetilde{G}^*$:

$$\begin{aligned} \max_{\widetilde{G} \subseteq G} \quad & \min_{i \in V(\widetilde{G})} \sum_{j \in \widetilde{N}(i)} d_{ij} \\ \text{s.t.} \quad & |V(\widetilde{G})| = m, \end{aligned} \quad (52)$$

where $\widetilde{N}(i)$ defines the neighbors of i in $\widetilde{G}$, i.e., for all $j \in \widetilde{N}(i)$ we have $j \in V(\widetilde{G})$ and $(i,j) \in E$.

- *Min-Diff DP*
  – Unweighted case: let the maximum degree of a node in graph G be denoted by $\Delta(G)$. Then problem (14) identifies a subgraph $\widetilde{G}^* \subseteq G$ with exactly m nodes that minimizes $\Delta(\cdot) - \delta(\cdot)$, i.e.,

$$\begin{aligned} \min_{\widetilde{G} \subseteq G} \quad & \left\{ \Delta(\widetilde{G}) - \delta(\widetilde{G}) \right\} \\ \text{s.t.} \quad & |V(\widetilde{G})| = m. \end{aligned} \quad (53)$$

We can also observe that if the solution of (53) is 0 then $\widetilde{G}^*$ is a *regular graph* (i.e., all vertices of $\widetilde{G}^*$ have the same degree, see, e.g., [10] for the definition of regular graphs). Therefore, the solution of (14) identifies the least irregular subgraph of G with exactly m nodes.

  – Weighted case: among all subgraphs of G with m nodes, we minimize the difference between the largest and smallest total weights of incidence sets in any such subgraph, i.e., we search for a subgraph $\widetilde{G}^* \subseteq G$ with m nodes where the total weights of edges incident to nodes in $\widetilde{G}^*$ are as close to each other as possible.

Furthermore, we note that the computational complexity results discussed in Section 5 can be also stated in the graph-theoretic context. For example, Proposition 6 directly implies the following corollary.

Corollary 1. *Given a weighted graph G, where edge weights can be both positive and negative, it is NP-hard to find a subgraph with the largest weighted density.*

Another, and probably the most interesting, observation can be made from Proposition 8. It is well-known that checking whether a given graph G contains a k-regular subgraph (i.e., subgraph with all vertices of degree k) is *NP*-complete for $k \geqslant 3$ [9]. Proposition 8 suggests an alternative view on the problem, which complements the already known complexity result.

Corollary 2. *The problem of checking whether a given graph G contains a regular subgraph with at least m nodes is NP-complete.*

Note that Corollary 2 does not contain any restrictions on regularity (e.g., the subgraph can be 1-regular, 2-regular, etc.), instead it requires the subgraph of certain size (at least m). Similarly, we can present a more general statement.

Corollary 3. *Given a graph G and two nonnegative integers m and r, the problem of checking whether G contains a subgraph $\widetilde{G}$ with at least m nodes such that $\Delta(\widetilde{G}) - \delta(\widetilde{G}) \leqslant r$ is NP-complete.*

7. Preliminary numerical studies

7.1. Test instances

We randomly generated 4 sets of inter-element distance matrices, $[d_{ij}]_{n \times n}$, given the number of elements $n = 30, 40, 50, 100$. We generated 10 fully-dense matrices in each matrix set and used uniform distribution $U(0, 200)$ to generate each d_{ij}. For each value n, we varied the cardinality value m and thus dealt with 10 instances given each pair of n and m. All computational experiments were performed on a PC with Intel Core 2 CPU of 2.66 GHz and RAM of 3 GB.

7.2. Exact MIP solutions

The first experiment is to compare the exact solution times for the linear mixed integer programming reformulations of the following four problems: *Maxsum DP*, *Maxmin DP*, *Max-Minsum DP*, and *Min-Diff DP*. All instances were solved using the CPLEX 9.0 MIP solver with default parameters. We also imposed a 1 hour time limit. The computational results are reported in Table 1.

7.3. Heuristic solution

The theoretical computational complexity results discussed in Section 5 indicate that the considered equitable dispersion problems are difficult combinatorial optimization problems. Therefore, heuristic based algorithms can be applied in order to obtain good-quality solutions in a fast manner. *Greedy Randomized Adaptive Search Procedure (GRASP)* is a rather popular tool for solving *Maxsum DP* and *Maxmin DP* (see, e.g., [15,31]). Motivated by the success of GRASP in solving *Maxsum DP* and *Maxmin DP*, we implemented a simple GRASP-based heuristic for solving *Max-Minsum DP*, which is a somewhat similar problem.

GRASP is a metaheuristic procedure, proposed by Feo and Resende [13]. The main components of GRASP are the construction and the improvement phases. In the construction phase, GRASP creates a complete solution by iteratively adding components of a solution with the help of a carefully designed greedy function, which is used to perform the selection. The improvement phase then takes the incumbent solution and performs local perturbations in order to get a local optimal solution, with respect to some predefined neighborhood. Different local search algorithms can be defined according to the chosen neighborhood. Next, we briefly describe the implementation details of the heuristic.

Construction Phase. Let M_k be a partial solution given by a set of $k(1 \leqslant k < m)$ elements selected, that is exactly k variables are fixed to be 1. The initial set M_1 is constructed randomly by selecting an index $i \in \{1,\ldots,n\}$. For any $i \in \{1,\ldots,n\} \setminus M_k$, we introduce $\Delta f^k(i)$ as the marginal contribution made by element i toward M_{k+1} from the current partial solution M_k. Let

$$s^k(i) = \sum_{j \in M_k} d_{ij}.$$

Then, for *Max-Minsum DP* we define

$$\Delta f^k(i) = \min\left\{\min_{j \in M_k}\{s^k(j) + d_{ji}\}, s^k(i)\right\} - f(M_k), \tag{54}$$

where $f(M_k)$ is the value of the objective function with variables from M_k fixed to 1.

Algorithm 1. Construction phase

input: coefficients d_{ji} for $j = 1,\ldots,n$ and $i = 1,\ldots,n$ and a constant n
output: a vector $x \in \{0,1\}^n$
$/^*$ initialize solution $x^*/$
$x \leftarrow (0,\ldots,0)$; $M_1 \leftarrow$ random index from {1, ...,n}
for $k \leftarrow 1$ *to* $m-1$ **do**
  $/^*$ create a restricted candidate list (RCL) $^*/$
  $L \leftarrow \{1,\ldots,n\} \setminus M_k$
  Order L according to function Δf^k as described in equation (54)
  RCL $\leftarrow$ first α elements of L
  Select random index $i \in$ RCL; $x_i \leftarrow 1$; $M_{k+1} \leftarrow M_k \cup \{i\}$
end
return x

The GRASP algorithm uses a list of candidate components, also known as the restricted candidate list (RCL). We sort the candidate variables according to their marginal contribution $\Delta f^k(i)$ to the objective function. Then we select the first α indices to form the RCL. At each iteration of the construction phase, parameter α is a random variable, uniformly distributed between 1 and $|L|$, the size of the set of variables to be fixed. One variable in the RCL is randomly selected at each iteration $k \geqslant 1$ with equal probability. It is then included into M_{k+1} and its value is set to 1. The implementation details of the construction phase are presented in Algorithm 1.

Improvement phase. The improvement phase of GRASP has the objective of finding a local optimal solution according to some neighborhood structure. The neighbors in our problem are obtained by perturbing the incumbent solution in the following way. Two different variables x_i and x_j are selected and their values are exchanged so that $\sum_{i=1}^n x_i$ remains unchanged. After the perturbation, the new solution is accepted if the objective is improved. Otherwise, another perturbation of the incumbent solution is performed. This phase terminates after N iterations without any

Table 1
Exact solution time comparison (10 instances of each size)

Problem size	Time\Measure	Maxsum	Maxmin	Max-Minsum	Min-Diff
$n = 30$	Maximum	360.8 second	1.594 second	0.547 second	41.50 second
$m = 10$	Median	237.3 second	0.953 second	0.352 second	34.14 second
	Minimum	105.6 second	0.672 second	0.094 second	22.06 second
$n = 30$	Maximum	327.3 second	14.72 second	1.016 second	56.63 second
$m = 15$	Median	232.1 second	5.969 second	0.602 second	42.53 second
	Minimum	80.26 second	3.141 second	0.218 second	25.89 second
$n = 40$	Maximum	$\geqslant$1 hour	5.078 second	2.453 second	822.5 second
$m = 10$	Median	$\geqslant$1 hour	3.704 second	1.430 second	682.0 second
	Minimum	$\geqslant$1 hour	2.781 second	0.578 second	535.8 second
$n = 40$	Maximum	$\geqslant$1 hour	86.91 second	7.594 second	3226 second
$m = 15$	Median	$\geqslant$1 hour	34.37 second	3.867 second	2678 second
	Minimum	$\geqslant$1 hour	10.09 second	0.500 second	1447 second
$n = 50$	Maximum	$\geqslant$1 hour	23.53 second	4.125 second	$\geqslant$1 hour
$m = 10$	Median	$\geqslant$1 hour	12.92 second	3.516 second	$\geqslant$1 hour
	Minimum	$\geqslant$1 hour	6.468 second	1.688 second	$\geqslant$1 hour
$n = 50$	Maximum	$\geqslant$1 hour	236.9 second	45.55 second	$\geqslant$1 hour
$m = 15$	Median	$\geqslant$1 hour	142.9 second	28.05 second	$\geqslant$1 hour
	Minimum	$\geqslant$1 hour	59.34 second	11.22 second	$\geqslant$1 hour
$n = 50$	Maximum	$\geqslant$1 hour	1638 second	180.4 second	$\geqslant$1 hour
$n = 20$	Median	$\geqslant$1 hour	592.9 second	145.6 second	$\geqslant$1 hour
	Minimum	$\geqslant$1 hour	241.4 second	25.06 second	$\geqslant$1 hour
$n = 50$	Maximum	$\geqslant$1 hour	$\geqslant$1 hour	481.0 second	$\geqslant$1 hour
$m = 25$	Median	$\geqslant$1 hour	3274 second	81.95 second	$\geqslant$1 hour
	Minimum	$\geqslant$1 hour	1376 second	10.73 second	$\geqslant$1 hour
$n = 50$	Maximum	$\geqslant$1 hour	$\geqslant$1 hour	112.3 second	$\geqslant$1 hour
$m = 30$	Median	$\geqslant$1 hour	$\geqslant$1 hour	32.22s	$\geqslant$1 hour
	Minimum	$\geqslant$1 hour	$\geqslant$1 hour	17.63 second	$\geqslant$1 hour
$n = 100$	Maximum	$\geqslant$1 hour	395.9 second	201.0 second	$\geqslant$1 hour
$m = 10$	Median	$\geqslant$1 hour	303.9 second	152.7 second	$\geqslant$1 hour
	Minimum	$\geqslant$1 hour	196.3 second	106.5 second	$\geqslant$1 hour
$n = 100$	Maximum	$\geqslant$1 hour	1967 second	1019 second	$\geqslant$1 hour
$m = 12$	Median	$\geqslant$1 hour	1288 second	799.0 second	$\geqslant$1 hour
	Minimum	$\geqslant$1 hour	778.5 second	350.0 second	$\geqslant$1 hour

Table 2
Computational results for solving Max-Minsum DP via GRASP

Problem size	Time	Exact solution	total_it = 500		total_it = 1000	
		CPU (s)	CPU (s)	Gap (%)	CPU(s)	Gap (%)
$n = 50$	Maximum	4.125	2.281	0.757	3.875	0.757
$m = 10$	Median	3.516	1.869	0.218	3.502	0.012
	Minimum	1.688	1.653	0.076	2.841	0
$n = 50$	Maximum	45.546	4.564	1.507	8.712	0.905
$m = 15$	Median	28.047	3.594	1.002	6.670	0.611
	Minimum	11.219	3.008	0.598	6.111	0.296
$n = 50$	Maximum	180.36	5.453	1.616	12.034	1.029
$m = 20$	Median	145.64	5.064	0.981	10.33	0.613
	Minimum	25.062	4.492	0.460	8.100	0.279
$n = 50$	Maximum	481.03	8.620	1.669	15.469	1.441
$m = 25$	Median	81.952	7.243	1.393	13.115	0.796
	Minimum	10.734	6.237	0.828	11.767	0.237
$n = 50$	Maximum	112.30	10.318	1.1554	14.045	0.902
$m = 30$	Median	32.219	8.874	0.623	13.118	0.408
	Minimum	17.625	6.278	0.301	4.748	0.085
$n = 100$	Maximum	201.02	6.466	3.718	6.355	4.034
$m = 10$	Median	152.72	5.561	3.185	5.814	2.957
	Minimum	106.52	5.045	2.335	5.228	2.493
$n = 100$	Maximum	1019.5	8.280	5.003	16.767	3.871
$m = 12$	Median	798.98	7.150	3.700	14.457	3.165
	Minimum	349.99	6.977	2.653	12.406	2.478

improvement, where N is a given parameter in the algorithm. In the computational experiments reported below, we set $N = 100$.

Termination criterion. The two-phase GRASP algorithm is set up to run until a fixed number of iterations *total_it* is reached.

Results. The computational results of GRASP on *Max-Minsum DP* are given in Table 2. Since GRASP is a randomized algorithm, we ran each instance 10 times and used the average heuristic suboptimal solution and CPU time to assess the performance.

7.4. Discussion

The first set of numerical experiments (see Table 1) was performed in order to compare relative difficulty of solving the MIP reformulations of two well-known measures (*Sum* and *Minimum* Dispersion) and two new measures proposed in the paper (*Minsum* and *Differential* Dispersion). The reported computational results in Table 1 indicate that *Maxsum DP* and *Min-Diff DP* are the most challenging problems; *Max-Minsum DP* seems to be the easiest one to solve among the presented measures; and *Maxmin DP* is somewhere in between. These results imply that in some situations we may prefer the selection of a particular measure due to requirements on the practical computational complexity of its respective linear mixed 0–1 programming formulation. The results may change if we vary the density of the inter-element distance matrices and/or the distribution used to generate the matrices. However, we can conclude that our MIP reformulations can handle problems only of relatively modest size (around 100 variables) in a reasonable amount of time on a standard PC.

Results in the second set of numerical experiments (see Table 2) show good performance of GRASP-based heuristics for solving *Max-Minsum DP*, which confirms similar results obtained for *Maxsum DP* and *Maxmin DP* reported in the literature [15,31]. The implemented heuristic is rather simple, however it was able to provide reasonably good solutions (about 3% optimality gap for largest problems) with a far better CPU time than the exact solution via CPLEX.

We also compared the solutions obtained via each measure. From our computational experiments, we conclude that the considered measures do not agree in general although some may match with each other more and some others less. This implies that the efficiency-based measures *sum* and *minimum* do not usually provide equitable dispersion among elements. Hence, to ensure an equitable system, we need to incorporate some equity-based measure. Our results also imply that to improve efficiency and, more importantly, equity, the measure selection must be based on the problem context as well as improvement requirements. Often times, more effort should be made in interpreting the equity results [36].

8. Conclusion

In this paper, we introduce three equity-based measures for the dispersion problem to address the imbalance of studying efficiency and equity in the optimization literature. We believe the main contribution of the paper is to enhance our understanding of equity in the optimization framework for the dispersion problems. We develop mathematical programs with proposed equity measures being objectives and present several mixed-integer linear reformulations. Computational complexity and graph-theoretic interpretation of these problems are studied.

The principles of the proposed equity measures can be applied in many other optimization problems where the issue of equity among various entities in the system must be addressed. As for solution, we may improve the mixed-integer reformulations and develop more efficient exact and heuristic algorithms.

After establishing these equity measures, it would be interesting to study multi-objective optimization that attempts to balance efficiency and equity. Efficiency and equity-based measures must be carefully selected [29]. In addition, the model should provide a flexible and interactive way for decision makers to make the tradeoff between efficiency and equity. It would also be interesting to apply stochastic and robust discrete optimization to address the equity issue under the uncertainty of problem parameters.

Acknowledgments

The authors would like to thank the anonymous referees for their helpful comments. The research of the first two authors is partially supported by the grant from AFOSR. The research of the first author is also supported by NSF grant CMMI-0825993.

References

[1] G.K. Adil, J.B. Ghosh, Maximum diversity/similarity models with extension to part grouping, International Transactions in Operations Research 12 (2005) 311–323.

[2] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice-Hall, Upper Saddle River, NJ, 1993.

[3] Ş. Ağca, B. Eksioglu, J.B. Ghosh, Lagrangian solution of maximum dispersion problems, Naval Research Logistics 47 (2000) 97–114.

[4] B. Bollobás, Modern Graph Theory, Springer, New York, NY, 1998.

[5] I.M. Bomze, M. Budinich, P.M. Pardalos, M. Pelillo, The maximum clique problem, in: D.-Z. Du, P.M. Pardalos (Eds.), Handbook of Combinatorial Optimization, vol. A, Kluwer Academic Publishers, Boston, MA, 1999, pp. 1–74.

[6] J.R. Brown, The knaspack sharing problem, Operations Research 27 (2) (1979) 341–355.

[7] J.R. Brown, The sharing problem, Operations Research 27 (2) (1979) 324–340.

[8] A. Caprara, D. Pisinger, P. Toth, Exact solution of the quadratic knapsack problem, INFORMS Journal on Computing 11 (1999) 125–137.

[9] F. Cheah, D.G. Corneil, The complexity of regular subgraph recognition, Discrete Applied Mathematics 27 (1990) 59–68.

[10] R. Diestel, Graph Theory, second ed., Springer, New York, NY, 2000.

[11] E. Erkut, S. Neuman, Analytical models for locating undesirable facilities, European Journal of Operational Research 40 (1989) 275–291.

[12] E. Erkut, The discrete *p*-dispersion problem, European Journal of Operational Research 46 (1990) 48–60.

[13] T.A. Feo, M.G.C. Resende, Greedy randomized adaptive search procedures, Journal of Global Optimization 6 (1995) 109–133.

[14] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, New York, NY, 1979.

[15] J.B. Ghosh, Computational aspects of the maximum diversity problem, Operations Research Letters 19 (1996) 175–181.

[16] F. Glover, C.-C. Kuo, K.S. Dhir, Heuristic algorithms for the maximum diversity problem, Journal of Information and Optimization Sciences 19 (1998) 109–132.

[17] P. Hansen, I.D. Moon, Dispersing facilities on a network, Cah CERO 36 (1994) 221–234.

[18] R. Hassin, S. Rubinstein, A. Tamir, Approximation algorithms for maximum dispersion, Operations Research Letters 21 (1997) 133–137.

[19] H. Huang, P. Pardalos, O.A. Prokopyev, Lower bound improvement and forcing rule for quadratic binary programming, Computational Optimization and Applications 33 (2006) 187–208.

[20] R.K. Kincard, Good solutions to discrete noxious location problems via metaheuristics, Annals of Operations Research 40 (1992) 265281.

[21] G. Kortsarz, D. Peleg, On choosing a dense subgraph, in: Proceedings of the 34th IEEE Symposium on Foundations of Computer Science, Palo Alto, CA, 1993, pp. 692–701.

[22] M.J. Kuby, Programming models for facility dispersion: The *p*-dispersion and maximum dispersion problems, Geographical Analysis 19 (1987) 315–329.

[23] C.-C. Kuo, F. Glover, K.S. Dhir, Analyzing and modeling the maximum diversity problem by zero-one programming, Decision Sciences 24 (1993) 1171–1185.

[24] I.D. Moon, S.S. Chaudhry, An analysis of network location problems with distance constraints, Management Science 30 (1984) 290–307.

[25] P.M. Pardalos, G.P. Rodgers, Computational aspects of a branch and bound algorithm for quadratic zero-one programming, Computing 45 (1990) 141–144.

[26] P.B. Mirchandani, R.L. Francis (Eds.), Discrete Location Theory, Wiley, Hoboken, NJ, 1990.

[27] J.-C. Picard, M. Queyranne, A network flow solution to some nonlinear 0–1 programming problems with applications to graph theory, Networks 12 (1982) 141–159.

[28] D. Pisinger, Upper bounds and exact algorithms for *p*-dispersion problems, Computers and Operations Research 33 (2006) 1380–1398.

[29] M. Rahman, M.J. Kuby, A multiobjective model for locating solid waste transfer facilities using an empirical opposition function, Information Systems and Operations Research 33 (1995) 3449.

[30] S.S. Ravi, D.J. Rosenkrantz, G.K. Tayi, Heuristic and special case algorithms for dispersion problems, Operations Research 42 (1994) 299310.

[31] M.G.C. Resende, R. Marti, M. Gallego, A. Duarte, GRASP and path relinking for the max–min diversity problem, Computers and Operations Research (2008), doi:10.1016/j.cor.2008.05.011.

[32] M. Tawarmalani, S. Ahmed, N.V. Sahinidis, Global optimization of 0–1 hyperbolic programs, Journal of Global Optimization 24 (2002) 385–416.

[33] M.B. Teitz, Toward a theory of public facility location, Papers in Regional Science 21 (1968) 35–51.

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

[35] T.-H. Wu, A note on a global approach for general 0–1 fractional programming, European Journal of Operational Research 101 (1997) 220–223.

[36] H.P. Young, Equity: In Theory and Practice, Princeton University Press, Princeton, NJ, 1993.