European Journal of Operational Research 000 (2016) 1–10

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Innovative Applications of O.R.

An integer programming approach for solving the *p*-dispersion problem

Fatemeh Sayyady[a,*], Yahya Fathi[b]

[a] SAS Institute, Inc., 100 SAS Campus Drive Cary, NC 27513, United States
[b] Industrial and Systems Engineering, North Carolina State University, Raleigh, NC 27695, United States

ARTICLE INFO

Article history:
Received 14 July 2013
Accepted 15 February 2016
Available online xxx

Keywords:
Location problem
Max–min diversity
Integer programming
Traffic sensor location

ABSTRACT

Given a collection of n items (elements) and an associated symmetric distance d_{ij} between each pair of items i and j, we seek a subset P of these items (with a given cardinality p) so that the minimum pairwise distance among the selected items is maximized. This problem is known as the *max–min diversity problem* or the *p-dispersion problem*, and it is shown to be *np*-hard. We define a collection of node packing problems associated with each instance of this problem and employ a binary search among these node packing problems to devise an effective procedure for solving the original problem. We employ existing integer programming techniques, i.e., branch-and-bound and strong valid inequalities, to solve these node packing problems. Through a computational experiment we show that this approach can be used to solve relatively large instances of the p-dispersion problem, i.e., instances with more than 1000 items. We also discuss an application of this problem in the context of locating traffic sensors in a highway network.

1. Introduction

We consider the problem of selecting a subset P (of size p) of items (elements) from a given collection N (of size n) so as to maximize diversity among the selected elements. Typically there is a characteristic vector $\mathbf{s}_i = (s_{i1}, s_{i2}, \ldots, s_{iK})$ associated with each element i in this collection, where s_{ik} represents the state or the value of the kth attribute of this element, and the difference between two elements i and j is defined as a normed distance d_{ij} between their corresponding characteristic vectors. For example, by a Euclidean distance measure we have $d_{ij} = \sqrt{\sum_{k=1}^{K} (s_{ik} - s_{jk})^2}$.

As stated in Kuo, Glover, and Dhir (1993) and (Resende, Marti, Gallego, & Duarte, 2010), there are basically two approaches to formulate this problem: the max-sum model and the max–min model. In the max-sum model the objective is to maximize the summation of all pairwise distances between the selected elements in the set P, while in the max–min model the objective is to maximize the minimum pairwise distance among these elements. Resende et al. (2010) refer to the former as the *maximum diversity problem* (*MDP*) and to the latter as the*max–min diversity problem* (*MMDP*). Both models have numerous applications and they have been the subject of several research articles in the past few decades. The former problem (*MDP*) has been studied in Kuby (1987); Kuo et al. (1993), and Pisinger (2006), while the latter (*MMDP*) is addressed in Erkut (1990); Kuby (1987), and Resende et al. (2010). Based on these studies it appears that from the point of view of their computational requirements *MDP* (max-sum) problem is somewhat easier to solve, but as argued in Kuo et al. (1993) the notion of diversity is better achieved through the max–min criterion (*MMDP*).

In this paper we address the max–min diversity problem (*MMDP*), which is also known as the p-dispersion problem (Erkut, 1990 and Resende et al., 2010). More specifically, for each subset P of N we define its associated *minimum dispersion*, denoted by $\Phi(P)$, as the smallest distance d_{ij} between any pair of elements i and j in this set, i.e., $\Phi(P) = \min_{\substack{i,j \in P \\ i \neq j}} \{d_{ij}\}$. The *p-dispersion problem* is then defined as the problem of finding a subset P, with a given cardinality p, that has the largest associated minimum dispersion $\Phi(P)$ among all such subsets.

The p-dispersion problem arises in a number of well documented applications in various fields. Moon and Chaudhry (1984) discuss its application in several network location problems. Erkut (1990) discusses its application in selecting the location of missile silos as well as in selecting locations for branches of a chain (franchise) so as to minimize mutual competition between similar shops or service stations. Pisinger (2006) discusses the application of this problem in telecommunications, such as in selecting

* Corresponding author. Tel.: +19196736088.
E-mail addresses: fatemeh.sayyady@gmail.com, fatemeh.sayyady@sas.com (F. Sayyady), fathi@ncsu.edu (Y. Fathi).

http://dx.doi.org/10.1016/j.ejor.2016.02.026
0377-2217/

locations for radio trans-receivers to service cellular phones in order to minimize interference. Kuo et al. (1993) discuss applications of this problem in several contexts such as in market planning (Keely & Ann, 1989), plant breeding (Porter, Rawal, Rachie, Wien, & William, 1975), social problems (Swierenga, 1977), and locating polluting industry (Dasarathy & White, 1980), among others.

We encountered this problem in the context of a traffic network where we wish to select a collection of locations to install weigh-in-motion sensors so as to obtain (collect) diverse information about the nature of traffic in the network. Due to budget constraints we limit the total number of sensors to p, and we wish to select p locations for these sensors (from among a given collection of n potential locations on the network) so as to maximize diversity in the collected traffic data. To this end, as discussed in Sayyady, Stone, List, Jadoun, and Kim (2013), maximizing the minimum *truck traffic dissimilarity* among the selected locations would be an appropriate objective function. We shall further discuss this application in Section 5.

In theory, solving the p-dispersion problem is relatively straightforward since it has only a finite number of feasible solutions, i.e., we can enumerate all possible subsets of p elements in the set N and select the subset that has the largest minimum dispersion. But the computational requirements of this approach can be prohibitively large even for relatively small instances of the problem, i.e., instances with relatively small values of n and p. Hence there is a need to develop computationally effective algorithms for solving the problem. As we discuss in the next section, several approaches for solving this problem are proposed in the open literature, and it is shown that these approaches can be employed to effectively solve relatively small to medium size instances of the problem (i.e., with $n \le 80$) within a reasonable execution time. But the computational requirements of these methods become excessive for larger instances of the problem.

In this paper we propose an alternative approach for solving the p -dispersion problem and show that this approach can be used to solve larger instances of the problem (e.g., with $n = 1000$) within a reasonable execution time. The approach that we employ is based on the strong relationship between the p-dispersion problem and a corresponding node packing problem (Nemhauser, George, & Wolsey, 1988), and we employ existing methods for solving the latter problem to solve the former.

In Section 2 we review the related literature and discuss several methods that are previously proposed for solving the p-dispersion problem. In Section 3 we discuss the relationship between the p-dispersion problem and a corresponding maximum cardinality node-packing problem. We then use this relationship to devise an exact procedure for solving the p-dispersion problem. In Section 4 we present the results of a computational experiment with the proposed method and show that the resulting procedure is indeed effective in solving relatively large instances of the problem. In Section 5 we discuss the application of this problem in the context of determining a set of locations for traffic sensors within the highway network in the state of North Carolina and present our findings. Section 6 contains a few concluding remarks.

2. Background

As stated above the p-dispersion problem is defined for a collection N of n elements and a given measure of distance d_{ij} between each pair of elements i and j in this set. We assume that this distance measure is symmetric, i.e., $d_{ij} = d_{ji}$ for all i and j, and we use D to represent the corresponding n by n symmetric matrix. The *p-dispersion problem* is now defined as the problem of finding a subset P of N, with a given cardinality p, that has the largest associated minimum dispersion $\Phi(P)$ among all such subsets. Input for this problem is the $n \times n$ matrix D and an integer $p < n$, and output is the set P .

Erkut (1990) and Ghosh (1996) show that the p -dispersion problem is *np*-hard. Erkut and Neuman (1989) also emphasized the complexity of the p-dispersion problem in their survey paper on location problems with maximization objectives. Hence the computational requirements of any algorithm for solving large instances of this problem is likely to be excessive. Yet, attempts have been made to design effective models and algorithms for solving moderate size instances of the problem. Also heuristic procedures with smaller computational requirements are developed to obtain near optimal solutions for larger instances of the problem. In this section we review these previous approaches and discuss their capabilities and limitations. We also introduce definitions and terminology that we employ in subsequent sections.

We start by presenting a mathematical programming model for the p -dispersion problem. Associated with each element $i \in N$ we define a decision variables $y_i = 1$ if element i is selected to be in the set P , and $y_i = 0$ otherwise. The p-dispersion problem can now be stated as the following mathematical programming model.

$$\text{Maximize} \quad r = Minimum_{i \neq j} \left\{ d_{ij} y_i y_j \right\} \qquad \text{(IQP)}$$

$$\text{Subject to} \quad \sum_{i \in N} y_i = p$$

$$y_i = 0 \text{ or } 1 \text{ for all } i \in N$$

Let r_p^{opt} denote the optimal value of the objective function for this model. Kuby (1987) writes this model as the following equivalent integer linear programming (ILP) model.

$$r_p^{opt} = \text{Maximize} \quad r \qquad \text{(ILP1)}$$

$$\text{Subject to} \quad r \le M(2 - y_i - y_j) + d_{ij} \quad \text{for all } i, j \in N, \ i < j$$

$$\sum_{i \in N} y_i = p$$

$$y_i = 0 \text{ or } 1 \text{ for all } i \in N$$

Our computational experiments show that this model can be employed to solve relatively small to medium size instances of the problem (with $n \le 100$) within a reasonable execution time (less than 90 minutes) using a general integer programming (IP) solver such as CPLEX (ILOG, 2011). But the computational requirements of this approach become excessive as the size of the problem grows larger. Kuo et al. (1993) offer a different ILP model for this problem which is somewhat more effective than the above model. But the computational requirements of solving this model also grow prohibitively large as the number of nodes reaches $n = 150$. In Section 4.1 we discuss the computational requirements associated with these two models in more detail.

Erkut (1990) mentions a strategy for solving the p-dispersion problem by using its relationship with the maximal clique problem and with a related IP model that he refers to as the *r-separation problem*. He then outlines a binary search strategy on the value of r in this context which is quite similar to the approach that we propose here, and it requires solving a sequence of ILP models for the r-separation problem for different values of r. Subsequently he discusses earlier attempts to solve this ILP model via standard branch-and-bound techniques that employ its LP relaxation, and reports disappointing computational results (Chaudhry, McCormick, & Moon, 1986). Erkut (1990) refrains from fully developing this approach. Instead, he proceeds to develop a separate branch-and-bound scheme for solving the p-dispersion problem and carries out a computational experiment to show its effectiveness. He reports a typical computation time of 14 seconds for an instance with $n = 30$, and the execution time rapidly grows to half an hour for an instance with $n = 40$. Erkut (1990) also develops a constructive (greedy) heuristic for solving larger instances of the problem.