Information Sciences 425 (2018) 1–17

Contents lists available at ScienceDirect

Information Sciences

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

A hybrid binary particle swarm optimization for the obnoxious *p*-median problem

Geng Lin[a,*], Jian Guan[b]

[a] Department of Mathematics, Minjiang University, Fuzhou, 350121, China
[b] Modern Educational Technology Center, Minjiang University, Fuzhou 350121, China

ARTICLE INFO

Article history:
Received 12 November 2016
Revised 16 August 2017
Accepted 7 October 2017
Available online 9 October 2017

Keywords:
Metaheuristics
Binary particle swarm optimization
Iterated greedy local search
Obnoxious *p*-median problem
Combinatorial optimization

ABSTRACT

The obnoxious *p*-median problem can be formulated as a constrained binary linear program. It is NP-hard, and has a lot of real world applications. In this paper, a hybrid binary particle swarm optimization is proposed to solve the obnoxious *p*-median problem. A new position updating rule is presented to inherit the good structure of previous high quality solutions. Furthermore, two tabu based mutation operators are used to avoid the premature convergence and guide the search to a promising area. A greedy repair procedure is developed to repair infeasible solutions. In addition, an iterated greedy local search procedure is utilized to enhance the exploitation ability. Extensive experiments are done on a set of 72 benchmark instances from the literature. Experimental results and comparisons with some existing algorithms demonstrate the effectiveness of the proposed algorithm. In particular, the proposed algorithm finds new best solutions for 15 instances. Compared with existing algorithms, the proposed algorithm is able to find better average objective function value in a short average computing time.

© 2017 Elsevier Inc. All rights reserved.

1. Introduction

1.1. *Obnoxious* p*-median problem*

The facility location problem is an important combinatorial optimization problem and has received much attention in the past decades [3,15]. The facilities could be either desirable or obnoxious (undesirable). A facility is called obnoxious when it is desired to locate it as far as possible from the customers. The obnoxious facilities, such as chemical industrial plants, electric power supplier networks, nuclear reactors, and radio or wireless stations, affect people as well as the environment. Although necessary for society, these facilities are undesirable and often dangerous to their surroundings. With the increasing environmental, healthy, and social impact, the obnoxious facility location problem gains more and more attention. Various models [1,9,12,32,36] have been presented in the literature for locating the obnoxious facilities.

In this paper we consider the obnoxious *p*-median (O*p*M) problem, which is an obnoxious facility location problem. It can be described as follows. Given a set of clients I, a set of facilities J, and the distance d_{ij} between the client $i \in I$ and the facility $j \in J$, the O*p*M problem seeks to determinate a subset of J with p facilities, such that the sum of the minimum distance between each client and the set of facilities is maximized. The O*p*M problem is proved to be NP-hard [36].

* Corresponding author.
E-mail address: lingeng413@163.com (G. Lin).

https://doi.org/10.1016/j.ins.2017.10.020
0020-0255/© 2017 Elsevier Inc. All rights reserved.

1.2. Motivations and contributions

There has been an increasing interest in the study of the O*p*M problem recently both in academic and engineering fields. An exact algorithm based on branch-and-cut [9] has been presented to solve the O*p*M problem. Solving large O*p*M instances to optimality using exact methods is not practical due to the significant time needed. For this reason, heuristic algorithms have been proposed to find approximate solutions in a reasonable running time, see Salhi for a critical overview on heuristics [34]. Belotti et al. [9] proposed an eXploring Tabu Search (XTS) to solve the O*p*M problem. Recently, Colmenar et al. [11] presented a greedy randomized adaptive search procedure (GRASP), which is composed of a greedy constructive algorithm, a local search, and a filtering mechanism, for finding approximate solutions. Experimental results have shown that the above two heuristics can find good quality solutions. However, the solution time of the above two existing heuristics is high. The average CPU time required by the XTS is 511.35 s whereas the GRASP with 1000 iterations and 5000 iterations consumed on average 237.34 and 749.09 s respectively. The main reasons for the expensive solution times are as follows: (1) The new starting solutions were generated by their greedy constructive algorithms without adopting the information of previous found local optima. Furthermore, two different iterations in GRASP may produce similar starting solutions in the construction phase. So, the algorithm required a large number of iterations (1000 or 5000 iterations) to find good quality solutions. (2) To make the search within the feasible solution space, the local search of GRASP employed a swap neighborhood. The size of the neighborhood is n^2, where n is the number of facilities. The size is very large, which makes the local search time consuming. Hence, it is worthy of developing an effective algorithm for finding competitive results in a short running time.

Particle swarm optimization (PSO) is a population based evolutionary algorithm which was originally proposed by Kennedy and Eberhart [24]. Many PSO variants have been proposed to solve various continuous optimization problems [18,19,22,37,40]. By redefining the position and velocity of the particle, Kennedy and Eberhart [25] also presented a binary version of PSO to deal with a discrete problem. In recent years, binary particle swarm optimization (BPSO) has been successfully applied to many optimization problems, such as resource allocation problem [14], multidimensional knapsack problem [6,20], feature selection [4], steiner minimal tree problem [28], and cloud computing [17,33]. Specially, BPSO using problem specific information to update position [29,31] has shown to be very effective and robust for solving hard optimization problems.

When applied to discrete optimization problems, there are three key issues that may affect the performance of BPSO. The first one is the position updating rule used in BPSO. The position updating rule is vitally important to BPSOs because it affects the search direction. Traditional BPSOs [25] used sigmoid functions to generated new positions. It is difficult to obtain good quality solutions. The second one is the diversification strategy. The mutation is one of the most popular diversification strategies. Various mutation operators have been proposed to prevent the premature convergence of BPSOs. However, few work has been devoted to mutation by taking both diversification and avoiding cycling into consideration. The third one is the effectiveness of the local search procedure. Most of computation time of a BPSO is usually spent on local search. The speed of any local search significantly depends on the size of the neighborhood [27]. It is essential to design a neighborhood with small size to reduce the computation time.

In this study, we focus on developing a fast hybrid binary particle swarm optimization (HBPSO) to the O*p*M problem for obtaining good quality solutions. We address the above three issues and the key contributions of this paper are described below:

1. We redefine the position updating rule according to the characteristic of the O*p*M problem. It is well known that the distances between high quality solutions for combinatorial optimization problems are often small [27,39]. For the reason, the proposed position updating rule utilizes the previous obtained local optima to generate new starting solutions (new positions). The newly obtained solutions inherit the good structure of high quality solutions. It reduces the computing time required by our local search procedure.
2. Two tabu based mutation operators are employed. They mark the changed facilities in a tabu list and prevents them from being changed again in the subsequent generations. First, mutation operators generate diversified solutions and guide the search to a promising area. Second, we use a short term tabu list to avoid cycling (i.e., generating too similar initial solutions in different generations) as this can avoid redundant work. GRASP uses a large number of iterations to find good quality solutions. Compared to GRASP, our proposed algorithm can find competitive results in 0.6n (less than 300) generations, where n is the number of facilities.
3. An iterated greedy local search procedure is presented to intensify the search. The existing local search methods are based on the swap neighborhood. Different from the existing local search methods for the O*p*M problem, our proposed local search procedure uses two consecutive operators (a removing operator and an adding operator) to guarantee the search within the feasible region. The size of the neighborhood is much smaller than the existing swap neighborhood. The time complexity of the proposed local search procedure is lower. Due to the fact that most of the computation time of an evolution algorithm is usually spent on local search [26], it dramatically speeds up the proposed algorithm.
4. The computational assessment on 72 benchmark instances from the literature indicates that our proposed algorithm can find good quality solutions in a short computing time. Experimental results and comparisons showed that our algorithm found new best solutions on 15 instances. In addition, the proposed algorithm can find better average objective function value in less time than existing algorithms.

Download English Version:

https://daneshyari.com/en/article/6857027

Download Persian Version:

https://daneshyari.com/article/6857027

Daneshyari.com