Knowledge-Based Systems 142 (2018) 95–107

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Solving maximum set k-covering problem by an adaptive binary particle swarm optimization method

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 3 April 2017
Revised 20 November 2017
Accepted 22 November 2017
Available online 23 November 2017

Keywords:
Metaheuristics
Particle swarm optimization
Local search
Maximum set k-covering problem
Combinatorial optimization

ABSTRACT

The maximum set k-covering problem (MKCP) consists in selecting a subset of k columns from a given set of n columns, in such a way that the number of rows covered by the selected columns is maximized. The problem is NP-hard and has lots of applications. In this paper, we propose an adaptive particle swarm optimization for solving the maximum set k-covering problem. The proposed algorithm uses a greedy constructive procedure to generate an initial swarm with good quality solutions. Based on the characteristic of the MKCP, an iterative local search procedure is developed to enhance the solution quality. Furthermore, a position updating procedure and a mutation procedure with adaptive mutation strength are employed to guide the search to a more promising area. These strategies achieve a good tradeoff between exploitation and exploration. Extensive evaluations on a set of benchmark instances show that the proposed algorithm performs significantly better than the existing heuristic for MKCP. In particular, it yields improved lower bounds for 96 out of 150 instances, and attains the previous best known results for remaining 54 instances. The key features of the proposed algorithm are analyzed to shed light on their influences on the performance of the proposed algorithm.

1. Introduction

The minimum set covering problem (MSCP) is one of the most widely studied NP-hard combinatorial optimization problems. Given m rows and n columns, and an $(m \times n)$ sparse matrix of zero-one elements a_{ij}, where $a_{ij} = 1$, if row i is covered by column j, and $a_{ij} = 0$, otherwise. The MSCP seeks to cover the rows using the minimum number of columns. The maximum set k-covering problem (MKCP) is to identify a subset S of a given cardinality k from n columns, such that the number of the rows covered by S is maximized. MKCP is seen as a generalized version of MSCP.

The MKCP is known to be NP-hard [1], and has lots of industrial engineering applications, such as the maximum covering location problem [2], crew scheduling in railway [3], cloud computing [4–6], multi-depot train diver scheduling [7], clustering [8], wireless sensor networks [9], etc.

Due to the computational complexity of MSCP, exact algorithm is not practical for large scale instances. For this reason, researchers make a lot of efforts on developing heuristics for obtaining good quality solutions. Many different kinds of heuristics have been proposed for solving MSCP, such as particle swarm optimization [10], ant colony optimization [11], tabu search [12], etc. However, these techniques can not be applied to solve MKCP directly. Two swapping-based one-pass streaming algorithm [13,14] and a greedy-based one-pass streaming algorithm (GOPS) [15] were proposed for solving the MKCP. To the best of our knowledge, there is only one stochastic heuristic [18], which is a restart local search algorithm (RNKC), to MKCP in the literature. Due to the computational challenge and application capability of MKCP, it is worthwhile to develop more heuristics for solving MKCP.

Particle swarm optimization (PSO) [19] is a relatively new evolutionary algorithm. It has been applied to lots of optimization problems, such as bi-level pricing problem [20], minimum labelling steiner tree problem, energy management, power economic dispatch problem, etc. The conventional PSO (CPSO) was originally developed for solving continuous optimization problem. Aiming to deal with discrete problems, Kennedy and Ebehert firstly proposed a discrete PSO (DPSO) [21]. After that, DPSO has been successfully applied to lots of different optimization problems [10,22–26]. Binary PSO (BPSO) is easy to implement and has been demonstrated strong efficacy in solving NP-hard optimization problems. All these successful applications for solving the challenging optimization problems motivate us to develop the employment of the BPSO to deal with the MKCP.

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

https://doi.org/10.1016/j.knosys.2017.11.028
0950-7051/

The MKCP can be formulated as constrained binary programming problem. When applied to MKCP, there are three main issues that may affect the performance of BPSO. The first is the constraint handing technique used in BPSO. The main challenge in application of BPSO to the constrained combinatorial optimization problem is optimizing the objective function while ensuring nonviolation of the constraints [27]. Most particle position updating methods of BPSOs to unconstrained binary optimization problems are blind to constraints. As a consequence, the new positions (solutions) generated by these techniques may be infeasible. The second is the effectiveness of the local search procedure. Local search procedure is often employed as an intensification strategy in evolutionary algorithms. Most of solution time of an evolutionary algorithm is usually spent on local search [28,29]. The third is a good balance between exploitation and exploration. Various exploration strategies [30–32] have been proposed to diversify the search.

In this paper, we take the characteristic of the MKCP to propose an adaptive binary particle swarm optimization (ABPSO) for obtaining high quality solutions in a short time. Three main contributions are made in this paper to efficiently solve the MKCP.

First, a new position updating rule is developed to generate new feasible solutions. It is commonly accepted that the degree of similarity between high quality solutions is generally very large for combinatorial optimization problems. The proposed position updating rule guides the search close to the previous found local optima (the personal best positions and the global best position). Moreover, to diversify the search, a particle is allowed to move close to other personal best positions.

Second, we present a modification of the Fiduccia–Mattheyses algorithm (FM) [33] for graph partitioning as local search procedure (denoted by MKCFM) to intensify the search. Based on the characteristic of the MKCP, the MKCFM uses two consecutive operators (removing operator and adding operator) to guarantee the search within the feasible region. It can find good solutions of MKCP quickly.

Third, the ABPSO employs an adaptive mutation operator to guide the search to a promising area. Each particle in the swarm is associated with a mutation strength variable. These variables are adjusted through self-adaption.

To assess the performance of the proposed algorithm in terms of both solution quality and solution time, we provide experimental results on a total of 150 benchmark instances from the literature, showing that the proposed algorithm can obtain high quality solutions in a short computing time. Moreover, it is able to find new improved solutions for 96 out of 150 instances, and attain the previous best known solutions for the other 54 instances.

The paper is organized as follows. Section 2 presents the problem description of the MKCP, and briefly introduces the existing algorithms. The proposed algorithm ABPSO is described in details in Section 3. Section 4 is dedicated to the computational results, and investigates two important features of the proposed algorithm. Concluding remarks are given in Section 5.

2. Mathematical model and related work

2.1. Problem formulation and notations

The MKCP can be formally defined as follows. Let $M = \{1, \cdots, m\}$ and $N = \{1, \cdots, n\}$ be the row set and column set, respectively. Let $A = (a_{ij})$ be an m-row, n-column, zero-one matrix. We say that a column j covers a row i if $a_{ij} = 1$. The MKCP consists in selecting a subset $S \subseteq N$ of cardinality k (hence $|S| = k$), such that the number of rows covered by S is maximized.

Let $x_j = 1$ for column $j \in S$ and $x_j = 0$ for column $j \in N - S$. The time of row i covered by S can be represented by

$$c_i = \Sigma_{j=1}^{n} a_{ij} x_j. \tag{1}$$

We use $\alpha_i \in \{0, 1\}$ to record whether the row i is covered by S. More formally,

$$\alpha_i = \begin{cases} 1 & if \quad c_i \geq 1, \\ 0 & otherwise, \end{cases}$$

and row i is covered by S if $\alpha_i = 1$. The MKCP can be formulated through a constrained 0–1 programming as follows:

$$\begin{cases} \max & f(x) = \sum_{i=1}^{m} \alpha_i, \\ s.t. & \sum_{j=1}^{n} x_j = k, \\ & x \in \{0, 1\}^n. \end{cases}$$

Let $I_j \subset M$ and J_i be the set of rows covered by column j, and the set of columns that are able to cover row i, respectively. More formally,

$$I_j = \{i \in M : a_{ij} = 1\}, J_i = \{j \in N : a_{ij} = 1\}.$$

We say column j_1 is adjacent to column j_2 if there exists a row $i \in M$ which is covered by columns j_1 and j_2, i.e., there exists a row i such that $j_1 \in J_i$ and $j_2 \in J_i$. Let AD_j be the set of columns which are adjacent to column j.

2.2. Previous work

The MKCP has been proved to be NP-hard. It can be approximated by an easy randomized method to under $(1 - \frac{1}{e}) \approx 0.632$ [15,18]. The guaranteed approximation ratios of existing approximation algorithms to MKCP are not satisfied for the good performance in recent applications.

In 2013, Yu and Yuan proposed a greedy-based one-pass streaming algorithm (GOPS) for solving the MKCP. Let C be a collection of checked columns. The GOPS started with $C = \emptyset$, and compared every column j to the columns in C. More formally, let $U_j = \{t : |I_t| > \frac{|I_j|}{\beta}, t \in C\}$, and $L_j = \{t : |I_t| \leq \frac{|I_j|}{\beta}, t \in C\}$, where $\beta > 0$ is a parameter. Then, the GOPS deleted the rows covered by the column j which are also covered by the columns in U_j. Afterwards, if the number of rows covered by the column j remains at least $\frac{|I_j|}{\beta}$, GOPS added the column j into C. Otherwise, as the number of rows covered by the column j decreases, GOPS updated U_j and kept on comparing the column j with the new U_j, until either the column j is added, or it was empty and discarded. If the column j is added into C, GOPS updated L_j by deleting all rows covered by the columns in L_j which are also covered by the column j. The GOPS executed by calling the above process to every column in a stream.

The GOPS is able to produce a prefix-optimal ordering of columns. Since the GOPS executed only one pass through the entire dataset, it is very fast. However, the solution quality produced by GOPS remains to be improved.

Heuristic algorithm has been shown to be an effective way to solve NP-hard optimization problems [16,17]. Recently, Wang et al. proposed a restart local search algorithm (RNKC) [18] for solving the MKCP. RNKC designed an initialization procedure to produce an initial solution. Then, a local search procedure was proposed to improve the initial solution. The above process was repeated until the predefined time limit is satisfied. Comprehensive results on a set of classical instances showed that RNKC competes very favorably with CPLEX.

3. The proposed algorithm ABPSO

In this section, we present the ABPSO algorithm to the MKCP. Firstly, we give the general framework of the proposed ABPSO. Then, several main components of ABPSO are described in detail.