Computers and Operations Research 93 (2018) 90–100

Contents lists available at ScienceDirect

Computers and Operations Research

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

Relaxation heuristics for the set multicover problem with generalized upper bound constraints[☆]

Shunji Umetani[a,*], Masanao Arakawa[b], Mutsunori Yagiura[c]

[a] Osaka University, Suita, Osaka, 565-0871, Japan
[b] Fujitsu Limited, Kawasaki 211-8588, Japan
[c] Nagoya University, Nagoya 464-8601, Japan

ARTICLE INFO

Article history:
Received 15 May 2017
Revised 5 January 2018
Accepted 6 January 2018
Available online 9 January 2018

Keywords:
combinatorial optimization
set covering problem
metaheuristics
local search
Lagrangian relaxation

ABSTRACT

We consider an extension of the set covering problem (SCP) introducing (i) multicover and (ii) generalized upper bound (GUB) constraints. For the conventional SCP, the pricing method has been introduced to reduce the size of instances, and several efficient heuristic algorithms based on such reduction techniques have been developed to solve large-scale instances. However, GUB constraints often make the pricing method less effective, because they often prevent solutions from containing highly evaluated variables together. To overcome this problem, we develop heuristic algorithms to reduce the size of instances, in which new evaluation schemes of variables are introduced taking account of GUB constraints. We also develop an efficient implementation of a 2-flip neighborhood local search algorithm that reduces the number of candidates in the neighborhood without sacrificing the solution quality. In order to guide the search to visit a wide variety of good solutions, we also introduce a path relinking method that generates new solutions by combining two or more solutions obtained so far. According to computational comparison on benchmark instances, the proposed method succeeds in selecting a small number of promising variables properly and performs quite effectively even for large-scale instances having hard GUB constraints.

1. Introduction

The set covering problem (SCP) is one of representative combinatorial optimization problems. We are given a set of m elements $i \in M = \{1, \dots, m\}$, n subsets $S_j \subseteq M$ ($|S_j| \geq 1$) and their costs c_j (> 0) for $j \in N = \{1, \dots, n\}$. We say that $X \subseteq N$ is a cover of M if $\cup_{j \in X} S_j = M$ holds. The goal of SCP is to find a minimum cost cover X of M. The SCP is formulated as a 0-1 integer programming (0-1 IP) problem as follows:

$$
\begin{array}{lll}
\text{minimize} & \sum_{j \in N} c_j x_j & \\
\text{subject to} & \sum_{j \in N} a_{ij} x_j \geq 1, & i \in M, \\
& x_j \in \{0, 1\}, & j \in N,
\end{array}
\quad (1)
$$

where $a_{ij} = 1$ if $i \in S_j$ holds and $a_{ij} = 0$ otherwise, and $x_j = 1$ if $j \in X$ and $x_j = 0$ otherwise. That is, a column $\boldsymbol{a}_j = (a_{1j}, \dots, a_{mj})^\top$ of matrix (a_{ij}) represents the corresponding subset S_j by $S_j = \{i \in M \mid a_{ij} = 1\}$. For notational convenience, for each $i \in M$, let $N_i = \{j \in N \mid a_{ij} = 1\}$ be the index set of subsets S_j that contains the element i.

The SCP is known to be NP-hard in the strong sense, and there is no polynomial time approximation scheme (PTAS) unless P = NP. However, the worst-case performance analysis does not necessarily reflect the experimental performance in practice. The continuous development of mathematical programming has much improved the performance of heuristic algorithms accompanied by advances in computing machinery (Caprara et al., 2000; Umetani and Yagiura, 2007). For example, Beasley (1990a) presented a number of greedy algorithms based on Lagrangian relaxation called the Lagrangian heuristics, and Caprara et al. (1999) introduced pricing techniques into a Lagrangian heuristic algorithm to reduce the size of instances. Several efficient heuristic algorithms based on Lagrangian heuristics have been developed to solve very large-scale instances with up to 5000 constraints and 1,000,000 variables with deviation within about 1% from the optimum in a reasonable computing time (Caprara et al., 1999; Caserta, 2007; Ceria et al., 1998; Yagiura et al., 2006).

☆ A preliminary version of this paper was presented in Umetani et al. (2013).

* Corresponding author. Tel.: +81-6-6879-7799.
E-mail addresses: umetani@ist.osaka-u.ac.jp (S. Umetani), arakawa.masanao@jp.fujitsu.com (M. Arakawa), yagiura@nagoya-u.jp (M. Yagiura).

https://doi.org/10.1016/j.cor.2018.01.007
0305-0548/

The SCP has important real applications such as crew scheduling (Caprara et al., 1999), vehicle routing (Hashimoto et al., 2009), facility location (Boros et al., 2005; Farahani et al., 2012), and logical analysis of data (Boros et al., 2000). However, it is often difficult to formulate problems in real applications as SCP, because they often have additional side constraints in practice. Most practitioners accordingly formulate them as general mixed integer programming (MIP) problems and apply general purpose solvers, which are usually less efficient compared with solvers specially tailored to SCP.

In this paper, we consider an extension of SCP introducing (i) multicover and (ii) generalized upper bound (GUB) constraints, which arise in many real applications of SCP such as vehicle routing (Bettinelli et al., 2014; Choi and Tcha, 2007), crew scheduling (Kohl and Karisch, 2004), staff scheduling (Caprara et al., 2003; Ikegami and Niwa, 2003) and logical analysis of data (Hammer and Bonates, 2006). The multicover constraint is a generalization of covering constraint (Pessoa et al., 2013; Vazirani, 2001), in which each element $i \in M$ must be covered at least $b_i \in \mathbb{Z}_+$ ($\mathbb{Z}_+$ is the set of nonnegative integers) times. The GUB constraint is defined as follows. We are given a partition $\{G_1, \ldots, G_k\}$ of N ($\forall h \neq h'$, $G_h \cap G_{h'} = \emptyset$, $\cup_{h=1}^{k} G_h = N$). For each block $G_h \subseteq N$ ($h \in K = \{1, \ldots, k\}$), the number of selected subsets S_j from the block (i.e., $j \in G_h$) is constrained to be at most d_h ($\leq |G_h|$). We call the resulting problem the *set multicover problem with GUB constraints* (SMCP-GUB), which is formulated as a 0-1 IP problem as follows:

$$\begin{array}{lll} \text{minimize} & z(\boldsymbol{x}) = \sum_{j \in N} c_j x_j & \\ \text{subject to} & \sum_{j \in N} a_{ij} x_j \geq b_i, & i \in M, \\ & \sum_{j \in G_h} x_j \leq d_h, & h \in K, \\ & x_j \in \{0, 1\}, & j \in N. \end{array} \tag{2}$$

This generalization of SCP substantially extends the variety of its applications. However, GUB constraints often make the pricing method less effective, because they often prevent solutions from containing highly evaluated variables together. To overcome this problem, we develop heuristic algorithms to reduce the size of instances, in which new evaluation schemes of variables are introduced taking account of GUB constraints. We also develop an efficient implementation of a 2-flip neighborhood local search algorithm that reduces the number of candidates in the neighborhood without sacrificing the solution quality. In order to guide the search to visit a wide variety of good solutions, we also introduce an evolutionary approach called the path relinking method (Glover and Laguna, 1997) that generates new solutions by combining two or more solutions obtained so far.

The SMCP-GUB is NP-hard, and the (supposedly) simpler problem of judging the existence of a feasible solution is NP-complete, since the satisfiability (SAT) problem can be reduced to this decision problem. We accordingly allow the search to visit infeasible solutions violating multicover constraints and evaluate their quality by the following penalized objective function. Note that throughout the remainder of the paper, we do not consider solutions that violate the GUB constraints, and the search only visits solutions that satisfy the GUB constraints. Let $\boldsymbol{w} = (w_1, \ldots, w_m) \in \mathbb{R}_+^m$ ($\mathbb{R}_+$ is the set of nonnegative real values) be a penalty weight vector. A solution $\boldsymbol{x}$ is evaluated by

$$\hat{z}(\boldsymbol{x}, \boldsymbol{w}) = \sum_{j \in N} c_j x_j + \sum_{i \in M} w_i \max\left\{ b_i - \sum_{j \in N} a_{ij} x_j, 0 \right\}. \tag{3}$$

If the penalty weights w_i are sufficiently large (e.g., $w_i > \Sigma_{j \in N} c_j$ holds for all $i \in M$), then we can conclude SMCP-GUB to be infeasible when an optimal solution $\boldsymbol{x}^*$ under the penalized objective function $\hat{z}(\boldsymbol{x}, \boldsymbol{w})$ violates at least one multicover constraint. In our algorithm, the initial penalty weights $\overline{w}_i$ ($i \in M$) are set to $\overline{w}_i = \sum_{j \in N} c_j + 1$ for all $i \in M$. Starting from the initial penalty weight vector $\boldsymbol{w} \leftarrow \overline{\boldsymbol{w}}$, the penalty weight vector $\boldsymbol{w}$ is adaptively controlled to guide the search to visit better solutions.

We present the outline of the proposed algorithm for SMCP-GUB. The first set of initial solutions are generated by applying a randomized greedy algorithm several times. The algorithm then solves a Lagrangian dual problem to obtain a near optimal Lagrangian multiplier vector $\tilde{\boldsymbol{u}}$ through a subgradient method (Section 2), which is applied only once in the entire algorithm. Then, the algorithm applies the following procedures in this order: (i) heuristic algorithms to reduce the size of instances (Section 5), (ii) a 2-flip neighborhood local search algorithm (Section 3), (iii) an adaptive control of penalty weights (Section 4), and (iv) a path relinking method to generate initial solutions (Section 6). These procedures are iteratively applied until a given time limit has run out.

2. Lagrangian relaxation and subgradient method

For a given vector $\boldsymbol{u} = (u_1, \ldots, u_m) \in \mathbb{R}_+^m$, called a Lagrangian multiplier vector, we consider the following Lagrangian relaxation problem LR($\boldsymbol{u}$) of SMCP-GUB:

$$\begin{array}{lll} \text{minimize} & z_{\text{LR}}(\boldsymbol{u}) = \sum_{j \in N} c_j x_j + \sum_{i \in M} u_i \left(b_i - \sum_{j \in N} a_{ij} x_j \right) & \\ & \quad = \sum_{j \in N} \left(c_j - \sum_{i \in M} a_{ij} u_i \right) x_j + \sum_{i \in M} b_i u_i & \\ \text{subject to} & \sum_{j \in G_h} x_j \leq d_h, \quad h \in K, & \\ & x_j \in \{0, 1\}, \quad j \in N. & \end{array} \tag{4}$$

We refer to $\tilde{c}_j(\boldsymbol{u}) = c_j - \sum_{i \in M} a_{ij} u_i$ as the Lagrangian cost associated with column $j \in N$. For any $\boldsymbol{u} \in \mathbb{R}_+^m$, $z_{\text{LR}}(\boldsymbol{u})$ gives a lower bound on the optimal value $z(\boldsymbol{x}^*)$ of SMCP-GUB (when it is feasible, i.e., there exists a feasible solution to SMCP-GUB).

The problem of finding a Lagrangian multiplier vector $\boldsymbol{u}$ that maximizes $z_{\text{LR}}(\boldsymbol{u})$ is called the Lagrangian dual problem (LRD):

$$\text{maximize} \left\{ z_{\text{LR}}(\boldsymbol{u}) \mid \boldsymbol{u} \in \mathbb{R}_+^m \right\}. \tag{5}$$

For a given $\boldsymbol{u} \in \mathbb{R}_+^m$, we can easily compute an optimal solution $\tilde{x}(\boldsymbol{u}) = (\tilde{x}_1(\boldsymbol{u}), \ldots, \tilde{x}_n(\boldsymbol{u}))$ to LR($\boldsymbol{u}$) as follows. For each block G_h ($h \in K$), if the number of columns $j \in G_h$ satisfying $\tilde{c}_j(\boldsymbol{u}) < 0$ is equal to d_h or less, then set $\tilde{x}_j(\boldsymbol{u}) \leftarrow 1$ for variables satisfying $\tilde{c}_j(\boldsymbol{u}) < 0$ and $\tilde{x}_j(\boldsymbol{u}) \leftarrow 0$ for the other variables; otherwise, set $\tilde{x}_j(\boldsymbol{u}) \leftarrow 1$ for variables with the d_h lowest Lagrangian costs $\tilde{c}_j(\boldsymbol{u})$ and $\tilde{x}_j(\boldsymbol{u}) \leftarrow 0$ for the other variables.

The Lagrangian relaxation problem LR($\boldsymbol{u}$) has integrality property. That is, an optimal solution to LR($\boldsymbol{u}$) is also optimal to its linear programming (LP) relaxation problem obtained by replacing $x_j \in \{0, 1\}$ in (4) with $0 \leq x_j \leq 1$ for all $j \in N$. In this case, any optimal solution $\boldsymbol{u}^*$ to the dual of the LP relaxation problem of SMCP-GUB is also optimal to LRD, and the optimal value z_{LP} of the LP relaxation problem of SMCP-GUB is equal to $z_{\text{LR}}(\boldsymbol{u}^*)$.

A common approach to compute a near optimal Lagrangian multiplier vector $\tilde{\boldsymbol{u}}$ is the subgradient method. It uses the subgradient vector $\boldsymbol{g}(\boldsymbol{u}) = (g_1(\boldsymbol{u}), \ldots, g_m(\boldsymbol{u})) \in \mathbb{R}^m$, associated with a given $\boldsymbol{u} \in \mathbb{R}_+^m$, defined by

$$g_i(\boldsymbol{u}) = b_i - \sum_{j \in N} a_{ij} \tilde{x}_j(\boldsymbol{u}). \tag{6}$$

This method generates a sequence of nonnegative Lagrangian multiplier vectors $\boldsymbol{u}^{(0)}, \boldsymbol{u}^{(1)}, \ldots,$ where $\boldsymbol{u}^{(0)}$ is a given initial vector and $\boldsymbol{u}^{(l+1)}$ is updated from $\boldsymbol{u}^{(l)}$ by the following formula:

$$u_i^{(l+1)} \leftarrow \max\left\{ u_i^{(l)} + \lambda \frac{\hat{z}(\boldsymbol{x}^*, \overline{\boldsymbol{w}}) - z_{\text{LR}}(\boldsymbol{u}^{(l)})}{\| \boldsymbol{g}(\boldsymbol{u}^{(l)}) \|^2} g_i(\boldsymbol{u}^{(l)}), 0 \right\}, \; i \in M, \tag{7}$$