European Journal of Operational Research 230 (2013) 212–225

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Advanced greedy algorithms and surrogate constraint methods for linear and quadratic knapsack and covering problems

Fred Glover

OptTek Systems, Inc., 2241 17th Street, Boulder, CO 80302, USA

ARTICLE INFO

Article history:
Received 8 January 2013
Accepted 7 April 2013
Available online 16 April 2013

Keywords:
Metaheuristics
Greedy algorithms
Knapsack/covering problems
Surrogate constraints
Multi-start/strategic oscillation
Tabu search

ABSTRACT

New variants of greedy algorithms, called *advanced greedy algorithms*, are identified for knapsack and covering problems with linear and quadratic objective functions. Beginning with single-constraint problems, we provide extensions for multiple knapsack and covering problems, in which objects must be allocated to different knapsacks and covers, and also for multi-constraint (multi-dimensional) knapsack and covering problems, in which the constraints are exploited by means of surrogate constraint strategies. In addition, we provide a new *graduated-probe strategy* for improving the selection of variables to be assigned values. Going beyond the greedy and advanced greedy frameworks, we describe ways to utilize these algorithms with multi-start and strategic oscillation metaheuristics. Finally, we identify how surrogate constraints can be utilized to produce inequalities that dominate those previously proposed and tested utilizing linear programming methods for solving multi-constraint knapsack problems, which are responsible for the current best methods for these problems. While we focus on 0–1 problems, our approaches can readily be adapted to handle variables with general upper bounds.

1. Introduction

Greedy algorithms have long been a mainstay of methods for single-constraint and multi-constraint knapsack and covering problems. These algorithms are often embedded within constructive processes used in multi-start metaheuristics and also within linked constructive and destructive processes in strategic oscillation metaheuristics. We introduce new variants of these algorithms, called *advanced greedy algorithms*, which can be implemented for problems with linear and quadratic objective functions.

In the quadratic case, we observe that previous formulations of greedy algorithms have conspicuous deficiencies and show how a new *graduated-probe strategy* can be used to overcome them both for greedy and advanced greedy methods. Though most strongly motivated in the quadratic setting, this new strategy can also be used to enhance variable-selection for problems with linear objectives. Starting with single-constraint problems, we then introduce extensions for more general multiple knapsack and multi-constraint (multi-dimensional) problems. Processes for employing these methods in strategic oscillation and multi-start approaches are also described.

In the domain of multi-constraint knapsack problems, we show how surrogate constraints can be used to provide inequalities that dominate inequalities used by previous methods incorporating linear programming strategies, which have produced the best existing methods for these problems. Our primary focus deals with 0–1 problems, though our methods can also be adapted for problems with more general integer variables.

Our paper is organized as follows. Section 2 begins by examining the simple case of single-constraint 0–1 knapsack and covering problems, including consideration of both linear and quadratic objectives. Section 3 describes classical greedy methods for these problems utilizing a framework that provides a foundation for later extensions. The new advanced greedy algorithms are introduced in Section 4, disclosing how these methods have the ability to eliminate certain deficiencies of the classical methods (and other previous methods). This section also identifies a way to produce fast updates for one of the main components of these methods.

Section 5 introduces the graduated-probe strategy and discloses the manner in which it overcomes limitations of methods proposed for quadratic problems. More general multiple knapsack problems, which have recently become the focus of a number of investigations, are addressed in Section 6. This section also describes how to reinforce such strategies in multi-start and strategic oscillation approaches. Finally, Section 7 discusses multi-constraint knapsack problems, together with the surrogate constraint strategies that generate inequalities to guide the solution process.

2. Single-constraint 0–1 knapsack and covering problems

Single-constraint 0–1 knapsack and covering problems with linear and quadratic objectives may be formulated as follows.

E-mail address: glover@opttek.com

0377-2217/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2013.04.010

Linear single-constraint knapsack (LK)

$$\text{Maximize } x_o = \sum(p_j x_j; j \in N) \tag{LK0}$$

$$\text{subject to } \sum(a_j x_j; j \in N) \leqslant a_o \tag{K1}$$
$$x_j \in \{0, 1\}, \quad j \in N \tag{K2}$$

where $a_o > 0$, and the profit coefficients p_j and the constraint coefficients a_j are likewise positive constants for all $j \in N = \{1, \ldots, n\}$. (Cases where some p_j or a_j coefficients are non-positive can easily be converted to the positive coefficient form by complementing variables or by observing that particular variables can be automatically assigned 0 or 1 values in an optimal solution.) We assume $\sum(a_j : j \in N) > a_o$, since $\sum(a_j : j \in N) \leqslant a_o$ implies (LK) has the trivial solution $x_j = 1$, all $j \in N$.

Linear single-constraint cover (LC)

$$\text{Minimize } y_o = \sum(c_j y_j; j \in N) \tag{LC0}$$

$$\text{subject to } \sum(d_j y_j; j \in N) \geqslant d_o \tag{C1}$$
$$y_j \in \{0, 1\}, \quad j \in N \tag{C2}$$

where similarly $d_o > 0$, and the cost coefficients c_j and the constraint coefficients d_j are positive constants for all $j \in N = \{1, \ldots, n\}$. (Cases where some c_j or d_j values are non-positive can likewise be converted to the positive coefficient form.) We assume $\sum(d_j; j \in N) > d_o$ since $\sum(d_j; j \in N) = d_o$ implies (LC) has the trivial solution $y_j = 1$, all $j \in N$ and $\sum(d_j; j \in N) < d_o$ implies (LC) has no feasible solution.

(LK) and (LC) are equivalent problems, as can be seen by replacing x_j by $1 - y_j$ in the former or replacing y_j by $1 - x_j$ in the latter and simplifying the resulting representation. (This produces constant terms in the objective functions for the two problems, but these do not affect the optimal solutions. The assumption $a_o > 0$ in (LK) corresponds to stipulating $\sum(d_j; j \in N) > d_o$ in (LC), and the assumption $d_o > 0$ in (LC) corresponds to stipulating $\sum(a_j; j \in N) > a_o$ in (LK).) In spite of this equivalence, however, the classical greedy algorithms for these problems do not yield equivalent solutions, and we treat them separately by indicating specific rules for each.

The quadratic versions of these problems arise by replacing the objective functions of the linear versions as follows:

Quadratic single-constraint knapsack (QK)

$$\text{Maximize } x_o = \sum(p_j x_j; j \in N) + \sum(p_{jh} x_j x_h : j, h \in N)$$
$$\text{subject to (K1) and (K2)} \tag{QK0}$$

Quadratic single-constraint cover (QC)

$$\text{Minimize } y_o = \sum(c_j y_j; j \in N) + \sum(c_{jh} y_j y_h : j, h \in N)$$
$$\text{subject to (C1) and (C2)} \tag{QC0}$$

In these formulations we assume for convenience that p_{jj} and c_{jj} are 0. This can be accomplished by setting $p_j := p_j + p_{jj}$ and $c_j := c_j + c_{jj}$, which is justified by the observation that a 0–1 variable z satisfies $z^2 = z$. We also assume p_{jh} and c_{jh} are 0 for $j > h$, in this case accomplished by setting $p_{jh} := p_{jh} + p_{hj}$ and $c_{jh} := c_{jh} + c_{hj}$ for $j < h$ (justified by the fact that $x_j x_h = x_h x_j$ and $y_j y_h = y_h y_j$). This latter assumption gives a means to save memory in storing data for the quadratic problems.

It is also customary to assume the profit coefficients p_j and p_{jh} and the cost coefficients c_j and c_{jh} are non-negative. However, this is not necessary for our development, though we retain the assumption that the a_j and d_j coefficients of (K1) and (K2) are positive. (The quadratic problems (QK) and (QC) are not equivalent under the assumption of non-negative p_{jh} and c_{jh} coefficients. In fact, equivalence between the two formulations requires $c_{jh} = -p_{jh}$.)

We begin by discussing the classical greedy algorithms as a prelude to introducing the advanced greedy algorithms.

3. Classical greedy algorithms for the single constraint problems

3.1. Algorithms for the linear objective single constraint problems

The classical greedy algorithms for (LK) and (LC) use the so-called bang-for-buck ratios $RK_j = p_j/a_j$ and $RC_j = c_j/d_j$ whose relevance for one-pass constructive methods was first noted in Dantzig (1957). Effectively, the algorithms go through the RK_j ratios in descending order and the RC_j ratios in ascending order, respectively, to assign $x_j = 1$ and $y_j = 1$ until no more assignments are possible that preserve feasibility for (LK) and until feasibility is achieved for (LC). We describe these algorithms in a form that is convenient for introducing later extensions, without consideration of details relating to efficient implementation (although we later describe fast updating methods for elements that change).

Greedy (LK)
Initialize:
$x_j = 0$ for all $j \in N$
$N1 = \varnothing$ ($N1$ is the index set for variables x_j currently set to 1)
RHS = a_o (RHS is the current "right hand side")
$N_o = \{j \in N: a_j \leqslant \text{RHS}\}$ (N_o is the index set for variables x_j that can feasibly be set to 1 at a given iteration of the following loop)
While RHS > 0 and $N_o \neq \varnothing$
$j^* = \arg\max(RK_j: j \in N_o)$
$x_{j^*} = 1$
$N1 = N1 \cup \{j^*\}$
$\text{RHS} = \text{RHS} - a_j^*$
$N_o = N_o - \{j^*\}$
$N_o = \{j \in N_o: a_j \leqslant \text{RHS}\}$
Endwhile
(Upon concluding, $x_j = 1$ for $j \in N1$ and $x_j = 0$ otherwise.)

We remark that the set N_o does not have to be explicitly identified and updated in the context of the simple (LK) problem, since it suffices to identify j^* by writing

$$j^* = \arg\ \max(RK_j : j \in N - N1 : a_j \leqslant \text{RHS})$$

However, explicit reference to N_o is useful for the context where knapsacks are generated by surrogate constraints in solving multidimensional knapsack problems, as discussed in Section 7.

Greedy (LC)
Initialize:
$y_j = 0$ for all $j \in N$
$N1 = \varnothing$ ($N1$ is the index set for variables y_j currently set to 1)
RHS = d_o (RHS is the current "right hand side")
While RHS > 0 and $N1 \neq N$
$j^* = \arg\min(RC_j: j \in N - N1)$
$y_{j^*} = 1$
$N1 = N1 \cup \{j^*\}$
$\text{RHS} = \text{RHS} - d_j^*$

(*continued on next page*)