Computers & Operations Research 40 (2013) 3100–3107

Contents lists available at ScienceDirect

Computers & Operations Research

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

Probabilistic GRASP-Tabu Search algorithms for the UBQP problem

Yang Wang [a], Zhipeng Lü [b], Fred Glover [c], Jin-Kao Hao [a,*]

[a] LERIA, Université d'Angers, 2 Boulevard Lavoisier, 49045 Angers, France
[b] School of Computer Science and Technology, Huazhong University of Science and Technology, 430074 Wuhan, China
[c] OptTek Systems, Inc., 2241 17th Street Boulder, CO 80302, USA

ARTICLE INFO

Available online 4 January 2012

Keywords:
GRASP
Tabu Search
UBQP
Path relinking
Population management
MaxCut

ABSTRACT

This paper presents two algorithms combining GRASP and Tabu Search for solving the Unconstrained Binary Quadratic Programming (UBQP) problem. We first propose a simple GRASP-Tabu Search algorithm working with a single solution and then reinforce it by introducing a population management strategy. Both algorithms are based on a dedicated randomized greedy construction heuristic and a tabu search procedure. We show extensive computational results on two sets of 31 large random UBQP instances and one set of 54 structured instances derived from the MaxCut problem. Comparisons with state-of-the-art algorithms demonstrate the efficacy of our proposed algorithms in terms of both solution quality and computational efficiency. It is noteworthy that the reinforced GRASP-Tabu Search algorithm is able to improve the previous best known results for 19 MaxCut instances.

1. Introduction

The objective of the unconstrained binary quadratic programming problem is to maximize the function:

$$f(x) = x'Qx = \sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij}x_i x_j \tag{1}$$

where $Q=(q_{ij})$ is an $n \times n$ matrix of constants and x is an n-vector of binary (zero-one) variables, i.e., $x_i \in \{0,1\}$, $i=1,\ldots,n$.

The UBQP is notable for its ability to formulate a wide range of important problems, including those from financial analysis [23], social psychology [16], machine scheduling [1], computer aided design [20] and cellular radio channel allocation [9]. Besides, due to the ability to incorporate quadratic infeasibility constraints into the objective function in an explicit manner, UBQP enables itself to serve as a common model for a wide range of combinatorial optimization problems. A review of additional applications and the re-formulation procedures can be found in Kochenberger et al. [19] demonstrating the utility of UBQP for a variety of applications.

During the last two decades, a large number of procedures for solving the UBQP have been reported in the literature. Among them are several exact methods using branch and bound or branch and cut (see, e.g., [6,17,30]). Due to the fact that the exact methods become prohibitively expensive to apply for solving large instances, various metaheuristic algorithms have been extensively used to find high-quality solutions to large UBQP instances in an acceptable time. Some representative metaheuristic methods include local search heuristics [7], Simulated Annealing [4,18], adaptive memory approaches based on Tabu Search [14,15,27,29]; population-based approaches such as Evolutionary Algorithms [5,21,25], Scatter Search [2] and Memetic Algorithms [22,26].

This paper presents two algorithms for the UBQP that combine GRASP and Tabu Search. The first, GRASP-TS, uses a basic GRASP algorithm with single solution search while the other, GRASP-TS/PM, launches each tabu search by introducing a population management strategy based on an elite reference set. In GRASP-TS/PM we pick a single solution at a time from the reference set, and operate on it, utilizing the ideas of "elite solution recovery" and "probabilistic evaluation" proposed in Glover et al. [12] and Xu et al. [37]. Our experimental testing discloses that GRASP-TS/PM yields very competitive outcomes on a large range of both random and structured problem instances.

To assess the performance and the competitiveness of our algorithms inn terms of both solution quality and efficiency, we provide computational results on 31 large random benchmark instances with up to 7000 variables as well as 54 instances derived from the MaxCut problem.

The remaining part of the paper is organized as follows. Sections 2 and 3 describe respectively the basic GRASP-Tabu Search algorithm and the GRASP-Tabu Search algorithm with Population Management. Section 4 is dedicated to the computational results and detailed comparisons with other

* Corresponding author.
E-mail addresses: yangw@info.univ-angers.fr (Y. Wang), zhipeng.lv@hust.edu.cn (Z. Lü), glover@opttek.com (F. Glover), hao@info.univ-angers.fr (J.K. Hao).

0305-0548/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2011.12.006

state-of-the-art algorithms in the literature. Finally, concluding remarks are given in Section 5.

2. GRASP-Tabu Search

2.1. General GRASP-TS procedure

The GRASP algorithm is usually implemented as a multistart procedure [31,32], consisting of a randomized greedy solution construction phase and a sequel local search phase to optimize the objective function as far as possible. These two phases are carried out iteratively until a stopping condition is satisfied.

Our basic GRASP-Tabu Search algorithm (denoted by GRASP-TS) for the UBQP follows this general scheme (see Algorithm 1) and uses a dedicated greedy heuristic for solution construction (see Section 2.2) as well as tabu search [13] (see Section 2.3) as its local optimizer.

Algorithm 1. Pseudo-code of GRASP-TS for UBQP.

```
Input: matrix Q
Output: the best binary n-vector x* found so far and its
objective value f*
f* = −∞
repeat
Construct a greedy randomized solution x^0 /* Section 2.2 */
    x' ← Tabu_Search(x^0)                  /* Section 2.3 */
    if f(x') > f* then
        x* = x', f* = f(x')
    end if
until a stopping criterion is met
```

2.2. Solution construction

GRASP-TS constructs a new solution at each step according to a greedy random construction heuristic, which was originally used in probabilistic Tabu Search (PTS) [12,36,37] and motivated by the fact that the GRASP construction resembles this PTS approach.

The construction procedure consists of two phases: one is to adaptively and iteratively select some variables to receive the value 1; the other is to assign the value 0 to the left variables. Starting with an empty solution, a variable x_i with the maximum q_{ii} is picked as the first element of the partial solution.

Given the partial solution $px = \{x_{k_1}, x_{k_2}, \ldots, x_{k_\alpha}\}$, indexed by $pi = \{k_1, k_2, \ldots, k_\alpha\}$, we calculate its objective function value (OFV) as

$$OFV(px) = \sum_{i \in pi, x_i = 1} \left(q_{ii} + \sum_{j \in pi, j \neq i} q_{ij} \cdot x_j \right) \tag{2}$$

At each iteration of the first phase we choose one unassigned variable according to a greedy function and then assign value 1 to it. Specifically, we calculate the objective function increment (OFI) to the partial solution px if one unassigned variable x_j ($j \in N \backslash pi$) is added into px with value 1

$$OFI_j(px) = q_{jj} + \sum_{i \in pi} (q_{ij} \cdot x_i) \tag{3}$$

At each step, all the unassigned variables are sorted in an non-increasing order according to their OFI values. Note that we only consider the first rcl variables having non-negative OFI values, where rcl is called the restricted candidate list size. The rth ranked variable is associated with a bias $b_r = 1/e^r$. Therefore, the rth ranked variable is selected with probability $p(r)$ that is calculated as follows:

$$p(r) = b_r \Big/ \sum_{j=1}^{rcl} b_j \tag{4}$$

Once a variable x_j is selected and added into the partial solution px with the assignment value 1, the partial solution px and its index pi, its objective function value $OFV(px)$ and the left variables' OFI values are updated correspondingly as follows:

$$px' = px \cup \{x_j\},\ pi' = pi \cup \{j\} \tag{5}$$

$$OFV(px') = OFV(px) + OFI_j(px) \tag{6}$$

For any variable x_k ($k \in N \backslash pi'$)

$$OFI_k(px') = OFI_k(px) + q_{jk} \tag{7}$$

This procedure repeats until all the OFI values of the unassigned variables are negative. Then, the new solution is completed by assigning the value 0 to all the left variables.

2.3. Tabu Search procedure

When a new solution is fully constructed, we apply the tabu search procedure described in Lü et al. [22] to optimize this solution. Our TS algorithm is based on a simple *one-flip move* neighborhood, which consists of changing (flipping) the value of a single variable x_i to its complementary value $1-x_i$. Each time a move is carried out, the reverse move is forbidden for the next *TabuTenure* iterations. In our implementation, we selected to set the tabu tenure by the assignment $TabuTenure(i) = ttc + rand(10)$, where ttc is a given constant and rand(10) takes a random value from 1 to 10. Once a move is performed, one needs just to update a subset of move values affected by the move. Accompanying this rule, a simple aspiration criterion is applied that permits a move to be selected in spite of being tabu if it leads to a solution better than the current best solution. Our TS method stops when the best solution cannot be improved within a given number μ of moves and we call this number the *improvement cutoff*. Interested readers are referred to [22] for more details.

3. GRASP-Tabu Search with population management

3.1. General GRASP-TS/PM procedure

Starting from the basic GRASP-TS algorithm, we introduce additional enhancements using the idea of maintaining a pool of elite solutions. By combining GRASP-TS with the population management strategy, our reinforced GRASP-TS/PM algorithm offers a better balance between intensification and diversification as a means of exploiting the search space. The general architecture of the GRASP-TS/PM algorithm is described in Algorithm 2, which is composed of four main components: a reference set construction procedure (lines 4, 23 in Algorithm 2, Section 3.2), a randomized greedy solution reconstruction operator (line 11 in Algorithm 2, Section 3.3), a tabu search procedure (line 12 in Algorithm 2, Section 2.3) and a reference set updating rule (lines 16–21 in Algorithm 2, Section 3.4).

GRASP-TS/PM starts from an initial reference set ($RefSet$) consisting of b local optimum solutions (line 4), from which the worst solution x^w in terms of the objective value is identified (line 6). Then, $Examine(x) = True$ indicates that solution x is to be examined (line 7). At each GRASP-TS/PM iteration, one solution x^0 is randomly chosen from the solutions to be examined in $RefSet$ ($Examine(x^0) = True$), reconstructed according to the randomized greedy heuristic and optimized by the tabu search procedure to