Computers & Operations Research 40 (2013) 166–179

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

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

Memetic search for the max-bisection problem

Qinghua Wu, Jin-Kao Hao*

LERIA, Université d'Angers 2 Boulevard Lavoisier, 49045 Angers Cedex 01, France

ARTICLE INFO

Available online 15 June 2012

Keywords:
Max-bisection
Max-cut
Memetic search
Evolutionary computing
Tabu search
Optimization

ABSTRACT

Given an undirected graph $G=(V,E)$ with weights on the edges, the max-bisection problem (MBP) is to find a partition of the vertex set V into two subsets V_1 and V_2 of equal cardinality such that the sum of the weights of the edges crossing V_1 and V_2 is maximized. Relaxing the equal cardinality, constraint leads to the max-cut problem (MCP). In this work, we present a memetic algorithm for MBP which integrates a grouping crossover operator and a tabu search optimization procedure. The proposed crossover operator preserves the largest common vertex groupings with respect to the parent solutions while controlling the distance between the offspring solution and its parents. Extensive experimental studies on 71 well-known G-set benchmark instances demonstrate that our memetic algorithm improves, in many cases, the current best known solutions for both MBP and MCP.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

Let $G=(V,E)$ be an undirected graph with vertex set $V=\{1,\dots,n\}$ and edge set $E \subset V \times V$, each edge $\{i,j\} \in E$ being associated with a weight $w_{ij} \in Z$. The well-known *max-cut* problem (MCP) is to partition the vertex set V into two disjoint subsets $V_1 \subset V$ and $V_2 = V \backslash V_1$ such that the sum of the weights of the edges from E that have one endpoint in each subset is maximized, i.e., $max\sum_{i \in V_1, j \in V_2} w_{ij}$. MCP is one of the first 21 NP-complete problems studied in [21]. When the two subsets V_1 and V_2 are required to have the same cardinality (assuming that n is even), the max-cut problem becomes the *max-bisection* problem (MBP) which remains NP-complete in the general case [12]. Both the max-bisection and max-cut problems have many applications such as statistical physics, classification, social network analysis, and VLSI design [5].

In this work, we are basically interested in the max-bisection problem. Given max-cut is a relaxed max-bisection problem, advances in solving max-bisection can benefit directly the solving of the max-cut problem.

There are two related 'dual' partition problems known as *minimum cut* and *minimum bisection* that aim to determine a two-way partition of a graph while minimizing the sum of the weights of the cutting edges (equal cardinality constraint is required for minimum bisection). Notice that in the general case, these minimization problems are different from the max-bisection and max-cut problems considered in this work which concern the two-way partition problems with the *maximization* criterion. Finally, graph partition problems with the minimization criterion have received much attention in the literature, leading to several well-known public-domain software packages like Chaco [18], Jostle [37], and Metis [22] (see [4] for a recent review).

The computational challenge of the general max-cut and max-bisection problems has motivated a variety of solution approaches including exact methods, approximation algorithms and metaheuristic methods. Examples of approximation algorithms based on semidefinite programming are described in [9,15,16,20,39]. These approaches provide a performance guarantee, but do not compete well with other methods in computational testing. Two recent examples of exact methods are described in [24,34] which are based on the cut and price approach and the branch and bound approach respectively. While these methods have the theoretical advantage of finding optimal solutions to a given problem, their applications are generally limited to problems with no more than a few hundred vertices.

For larger problem instances, a number of heuristics and metaheuristics are often used to find approximate solutions of good quality with a reasonable computing time. This includes for the max-bisection problem deterministic annealing [6], Lagrangian net [38] and variable neighborhood search [25]. There are many heuristics and metaheuristics for the max-cut problem including simulated annealing [1], rank-2 relaxation heuristic [5], GRASP [8], diversification driven tabu search [23], advanced scatter search [28], global equilibrium search [35], and probabilistic tabu search [36].

* Corresponding author.
E-mail addresses: wu@info.univ-angers.fr (Q. Wu), hao@info.univ-angers.fr (J.-K. Hao).

0305-0548/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2012.06.001

In this paper, we present a memetic algorithm for the max-bisection problem (denoted by MAMBP). The proposed algorithm integrates three complementary key components which jointly ensure the high efficiency of the search process. First, to generate promising new solutions, we introduce a dedicated crossover operator which tries to preserve groups of vertices that are shared by parent solutions. The design of this crossover operator relies on the observation that, given a set of high quality bisections of a graph, there is always a large number of vertices grouped together throughout these bisections. Second, we devise a tabu search optimization procedure for the purpose of intensified search around a given solution. The tabu search procedure uses a vertex move neighborhood and incremental evaluation techniques for a fast neighborhood examination. Finally, to maintain a healthy diversity of the population, we employ a pool updating strategy which takes into account both the solution quality and the distance between solutions.

We show extensive experimental results on 71 well-known G-set benchmark graphs (with 800 to 20000 vertices) in the literature, showing that the proposed algorithm achieves highly competitive results with respect to the existing max-bisection heuristics. Moreover, when considering the relaxed max-cut problem, the results produced by our MAMBP algorithm remain highly competitive even when they are compared to those obtained by dedicated max-cut algorithms; for 31 max-cut instances, MAMBP improves the previous best known max-cut solutions of the literature.

In the next section, the components of our memetic algorithm are described, including the tabu search procedure, the crossover operator and the pool replacement strategy. Section 3 is dedicated to computational results and detailed comparisons with other state-of-the-art algorithms in the literature. Section 4 investigates several essential parts of the proposed memetic algorithm, followed by concluding remarks given in Section 5.

2. Memetic algorithm

Memetic algorithms are known to be an effective approach in solving a number of hard combinatorial optimization problems [29,30,17]. Typically, a memetic approach repeatedly alternates between a recombination (or crossover) operator to generate solutions located in promising regions in the search space and a local optimization procedure to search around the newly generated solutions. It is commonly admitted that the success of this approach depends critically on the recombination operator. In order to be effective, the recombination operator must be adapted to the problem being solved and should be able to transmit meaningful features from parents to offspring.

The general scheme of our memetic approach for MBP is summarized in Algorithm 1. Basically, our memetic algorithm begins with an initial population of solutions which are first improved by the local optimization procedure based on tabu search [14] (lines 1–5, Sections 2.2 and 2.3) and then repeats an iterative process for a fixed number of times (generations) (lines 6–13). At each generation, two solutions are selected to serve as parents (Section 2.4). The crossover operator is applied to the parents to generate a new offspring solution (Section 2.5) which is further improved by the tabu search optimization procedure (Section 2.3). Finally, we apply a quality-and-diversity based rule to decide whether the improved offspring solution can be inserted into the population (Section 2.6). In the following subsections, we give more details on the components of our memetic algorithm.

Algorithm 1. Memetic algorithm for the max-bisection problem.

Require: A weighted graph $G=(V,E,\omega)$, population size p
Ensure: The best solution I^* found
1: $Pop=\{I_1,\ldots,I_p\}\leftarrow Initial\_Population()$
2: $I^*\leftarrow Best(Pop)$
3: **for** $i=1$ to p **do**
4: $I_i\leftarrow Tabu\_Search(I_i)$ /* Section 2.3 */
5: **end for**
6: **while** the stop criterion is not met **do**
7: Select randomly two solutions (parents) I_i and I_j from Pop /* Section 2.4 */
8: $I_0=Cross\_Over(I_i,I_j)$ /* Section 2.5 */
9: $I_0\leftarrow Tabu\_Search(I_0)$ /* Section 2.3 */
10: **if** $f(I_0)>f(I^*)$ **then**
11: $I^*\leftarrow I_0$ /* Update the best solution found so far */
12: **end if**
13: $Pop\leftarrow Pool\_Updating(I_0,Pop)$ /* Section 2.6 */
14: **end while**

2.1. Search space and cost function

Recall that MBP consists of partitioning the vertex set V into two subsets of equal cardinality such that the weights on the edges between the two subsets are maximized. As such, we define the search space explored by our memetic algorithm as the set of all possible partitions of V into two disjoint subsets of equal cardinality (also called bisections), i.e., $\Omega=\{\{V_1,V_2\}:|V_1|=|V_2|=|V|/2, V_1\cap V_2=\emptyset\}$. Clearly, the size of Ω is given by $C(|V|,|V|/2)$.

Given a bisection $I=\{V_1,V_2\}\in\Omega$, the cost function (also called the fitness function) $f(I)$ sums up the weights of the edges between the two subsets V_1 and V_2 such that:

$$f(I)=\sum_{i\in V_1, j\in V_2} w_{ij} \tag{1}$$

Then, for two bisections $I_A\in\Omega$ and $I_B\in\Omega$, I_A is better than I_B if and only if $f(I_A)>f(I_B)$. The goal of the max-bisection problem is to find:

$$\arg\max_{I\in\Omega} f(I)$$

Given the size of the search space Ω, it is particularly challenging to find an exact solution or an approximate solution of high quality.

2.2. Initial population

The solutions (individuals) of the initial population are created as follows. For each individual, an equal sized partition is first created at random and then improved by the tabu search procedure (see Section 2.3). The improved solution is added into the population if this solution is not already present in it. Otherwise, this solution is discarded and a new random (equal sized) partition is created. This procedure is iterated until the population is filled with p solutions (p is the population size). This simple procedure provides an initial population of diverse solutions of good quality.

2.3. The perturbation-based tabu search procedure

Our tabu search (TS) procedure aims to improve a given solution I and plays the key role of local optimization within our memetic algorithm. Basically, our tabu search procedure repeatedly alternates between an intensification phase ensured by the basic tabu search and a diversification phase controlled by a perturbation mechanism [14]. Algorithm 2 describes this perturbation-based tabu search procedure, whose components are detailed in the following subsections.

Download English Version:

https://daneshyari.com/en/article/10347490

Download Persian Version:

https://daneshyari.com/article/10347490

Daneshyari.com