

An Optimal Algorithm to Find a Maximum Weight 2-Coloured Set on Cactus Graphs

Kalyani Das¹ and Madhumangal Pal^{2, +}

¹Department of Mathematics, Ramnagar College, Ramnagar, Purba Medinipur--721481, India.

²Department of Applied Mathematics with Oceanology and Computer Programming, Vidyasagar University, Midnapore -- 721102, India.

(Received August 2, 2010, accepted August 8, 2010)

Abstract. A cactus graph is a connected graph in which every block is either an edge or a cycle. An optimal algorithm is presented here to find a maximum weight 2-coloured set on cactus graphs in $O(n)$ time, where n is the total number of vertices of the graph. The cactus graph has many applications in real life problems, specially in radio communication system.

Keywords: Design of algorithms, analysis of algorithms, 2-colour set, cactus graph.

1. Introduction

Let $G = (V, E)$ be a finite, connected, undirected, simple graph of n vertices and m edges, where V is the set of vertices and E is the set of edges. A vertex v is called a *cut-vertex* if removal of v and all edges incident to v disconnect the graph. A *non-separable graph* is a connected graph which has no cut-vertex and a *block* means a maximum non-separable sub-graph. A block is a *cyclic block* or simply *cycle* in which every vertex is of degree two.

A *cactus graph* is a connected graph in which every block is either an edge or a cycle.

The Graph Colouring Problem (GCP) plays a central role in graph theory and it has direct applications in real life problems [2, 4, 22], and is related to many other problems such as timetabling [11, 23, 27], frequency assignment [14] etc. A K -colouring (assignment) of an undirected graph $G = (V, E)$, where V is the set of $|V| = n$ vertices and $E \subseteq V \times V$ the set of edges, is a mapping $\alpha : V \rightarrow \{1, 2, \dots, K\}$ that assigns a positive integer from $\{1, 2, \dots, K\}$ (representing the colours) to each vertex. We say that a colouring is feasible if the end nodes of every edge in E have assigned different colours, i.e., for all $(u, v) \in E$, $\alpha(u) \neq \alpha(v)$. We call conflict the situation when two nodes between which an edge exists have the same colour associated to them. We say that a colouring is infeasible if at least one conflict occurs. Alternatively to the formulation as an assignment problem, the GCP can also be represented as a partitioning problem, in which a feasible K -colouring corresponds to a partition of the set of nodes into K sets $C_1, C_2, \dots, C_K$ such that no edge exists between two nodes from the same colour class.

The graph colouring problem is NP-complete. Hence, we need to use approximate algorithmic methods to obtain solutions close to the absolute minimum in a reasonable execution time.

The Maximum Weight k -Colourable Subgraph (MWKC) problem is related to the following problem. The input to this problem consist of an integer number k and an undirected graph $G = (V, E)$, where each vertex v has a non-negative weight w_v . The goal is to pick a subset $V' \subseteq V$, such that there exists a colouring c of $G[V']$ with k colours, and among all such subsets, the value $\sum_{v \in V'} w_v$ is maximum. This problem is NP-hard for general graph even for split graph [16].

The maximum weight k -colouring problem is same as the maximum weight k -independent set (MWKIS)

⁺ E-mail address: mmpalvu@gmail.com

problem. The maximum k -independent set problem on G is to determine k disjoint independent sets $S_1, S_2, \dots, S_k$ in G such that $S_1 \cup S_2 \cup \dots \cup S_k$ is maximum. The MWKIS problem is NP-complete for general graphs [16].

Many work on Colouring Problem has been done previously. Local search in large neighbour and iterated local search for GCP are described in [10, 1]. In [3] there are different new methods to colour the vertices of the graph. Vertex colouring by multistage branch and bound method is described in [7] and by branch and cut algorithm is described in [12]. Greedy graph colouring is used in [5]. Genetic and hybrid algorithm for graph colouring is applied in [13, 15]. In [22] parallel graph colouring is applied. Graph colouring algorithm for assignment problem in radio network is done in [8] and ant algorithm for GCP is described in [9].

The maximum weight 2-colouring problem or the maximum weight 2-independent set (MW2IS) problem, which is a special case of the (MWKIS) problem, is also NP-complete for general graphs [26] and its applications have been studied in the last decade [17, 18, 21, 26]. In [17], Hsiao et.al. have solved the two-track assignment problem by solving the M2IS problem on circular arc graph. In [21], Lou et. al. have solved the maximum 2-chain problem on a given point set, which is the same as the MW2IS problem on permutation graph.

Cactus graph has many applications. These graphs can be used to model physical setting where a tree would be inappropriate. Examples of such setting arise in telecommunications when considering feeder for rural, suburban and light urban regions [19] and in material handling network when automated guided vehicles are used in [20]. Moreover ring and bus structures are often used in local area networks. The combination of local area network forms a cactus graph.

To illustrate the problem we consider a weighted cactus graph of Figure 1. The numbers in the bracket of each vertices represent the weights of that vertices.

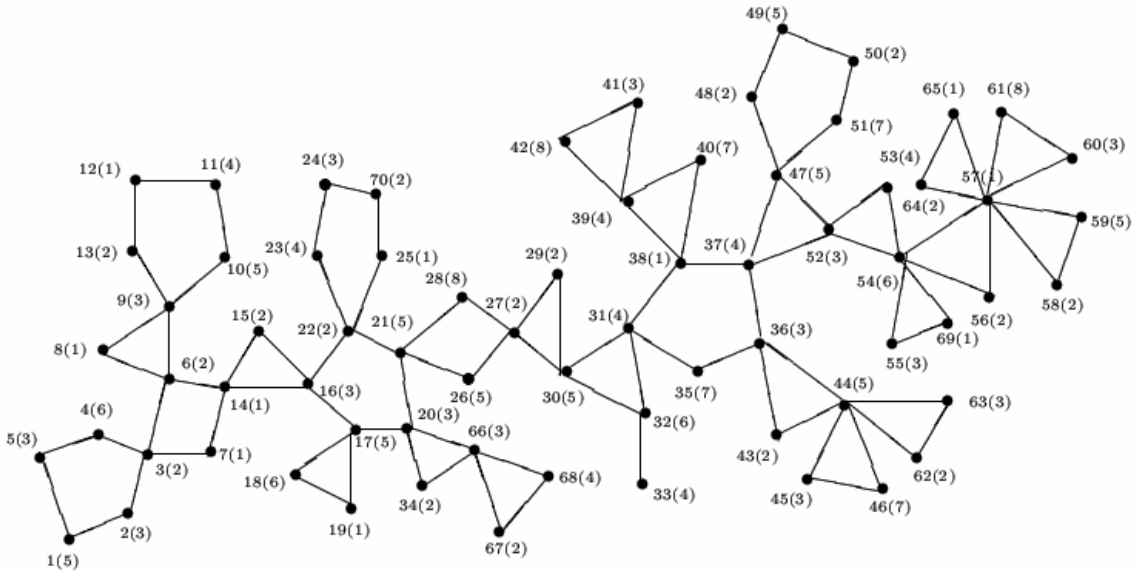


Figure 1: A cactus graph G .

In this paper maximum weight 2-colouring problem is considered on a weighted cactus.

2. Computation of blocks and cutvertices of G

As described in [25] the blocks as well as cut vertices of a graph G can be determined by applying DFS technique. Using this technique we obtain all blocks and cut vertices of the cactus graph $G = (V, E)$. Let the blocks be $B_1, B_2, B_3, \dots, B_N$ and the cut vertices be $c_1, c_2, c_3, \dots, c_R$ where N is the total number of blocks and R is the total number of cut vertices.

The blocks and cut vertices of the cactus graph shown in Figure 1 are respectively $B_1 = (1,2,3,4,5)$, $B_2 = (3,6,7,14)$, $B_3 = (6,8,9)$, $B_4 = (9,10,11,12,13)$, $B_5 = (14,15,16)$, $B_6 = (16,17,21,20,22)$,