Swarm and Evolutionary Computation xxx (2018) 1–6

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

A comparison study of harmony search and genetic algorithm for the max-cut problem

Yong-Hyuk Kim [a], Yourim Yoon [b,*], Zong Woo Geem [c]

[a] Department of Computer Science & Engineering, Kwangwoon University, 20 Kwangwoon-ro, Nowon-gu, Seoul, 139-701, Republic of Korea
[b] Department of Computer Engineering, Gachon University, 1342 Seongnamdaero, Sujeong-gu, Seongnam-si, Gyeonggi-do, 461-701, Republic of Korea
[c] Department of Energy & Information Technology, Gachon University, 1342 SeongnamDaero, Sujeong-Gu, Seongman-Si, Gyeonggi-do, 461-701, Republic of Korea

A B S T R A C T

The max-cut problem is one of well-known NP-complete problems and has applications in various fields such as the design process for VLSI (Very-Large-Scale Integration) chips and spin glass theory in statistical physics. In this paper, a harmony search algorithm for the max-cut problem is proposed. Compared to genetic algorithm, harmony search algorithm has advantages of generating a new vector after considering all of the existing vectors and requiring only a few number of parameters to be determined before the run of the algorithm. For 31 benchmark graphs of various types, the proposed harmony search algorithm is compared with a newly developed genetic algorithm and another genetic algorithm taken from the recent literature are presented. The proposed harmony search algorithm produced significantly better results than the two genetic algorithms.

1. Introduction

Many procedures for solving combinatorial optimization problems involve partitioning the vertices of a graph into two disjoint subsets of as near equal size as possible, such that the sum of the weights of the edges connecting vertices in different subsets is maximized or minimized. Given an undirected graph $G = (V, E)$ with weighted edges, the max-cut problem is that of finding a subset $S \subset V$ which maximizes the cut size, which is the sum of the weights of the edges which cross the cut separating S from $V - S$.

The max-cut problem has many applications. For example, the assignment of layers in the design of VLSI (Very-Large-Scale Integration) chips and PCBs (Printed Circuit Boards) can be reduced to a max-cut problem. The reduction by Pinter and by Chen, Kajitani and Chan, independently, is briefly introduced here [1,2]. For example case in Fig. 1, all cells are placed on a chip and all nets have been routed but that the assignment of wire segments to layers has not been performed. Only wires belonging to different nets may cross. So, a feasible layer assignment must have the property that such crossing wire segments are assigned to different layers. Physically, a change of layers on the same wire is achieved by placing a "via", which is a hole to be drilled. In printed circuit board fabrication, vias cause additional costly work, so it is desirable to find a layer assignment such that the number of vias is as small as possible. However, by several important design rules, placement of vias are restricted. The segments on which vias are allowed are drawn dotted in Fig. 1. These segments are called free segments. And critical segments, on which vias are forbidden, are drawn solid and numbered from 1 to 14 in the figure.

We want to treat the case where only two layers are available. The critical segments correspond to the node set V of a layout graph $G = (V, E)$ which has two kinds of edges. Nodes i and j are joined by a conflict edge ij whenever the associated critical segments cross each other. Nodes i and j are joined by a continuation edge ij whenever the associated critical segments are connected by a free segment. In our example, the layout graph looks as shown in Fig. 2. The conflict edges are drawn solid and the continuation edges drawn dotted. For the minimization of vias, we should find a cut C of G that contains all conflict edges and contains as few continuation edges as possible. This problem can be converted to the max-cut problem with a reduced graph from G. For constructing the reduced graph H, we partition the "conflict graph", which consists of V and conflict edges, into connected components. Each component is supposed to be bipartite, because if it is not, no feasible assignment of segments to two layers will exist. In our example, there are 4 connected bipartite components and those are shown in Fig. 3. For each component, a "representative" node is arbitrarily chosen and two nodes of the reduced graph are joined by an edge if

* Corresponding author.
E-mail addresses: yhdfly@kw.ac.kr (Y.-H. Kim), yryoon@gachon.ac.kr (Y. Yoon), geem@gachon.ac.kr (Z.W. Geem).

https://doi.org/10.1016/j.swevo.2018.01.004
Received 12 August 2017; Received in revised form 15 November 2017; Accepted 9 January 2018
Available online XXX

Please cite this article in press as: Y.-H. Kim, et al., A comparison study of harmony search and genetic algorithm for the max-cut problem, Swarm and Evolutionary Computation (2018), https://doi.org/10.1016/j.swevo.2018.01.004

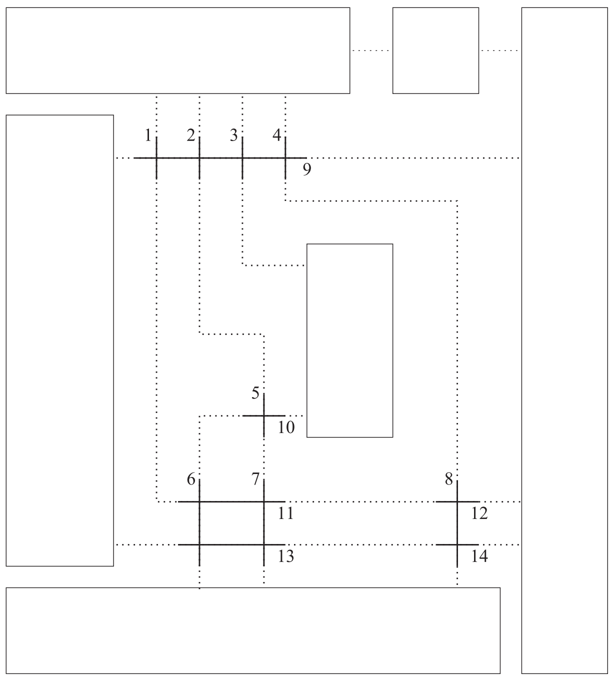

Fig. 1. An example VLSI chip.

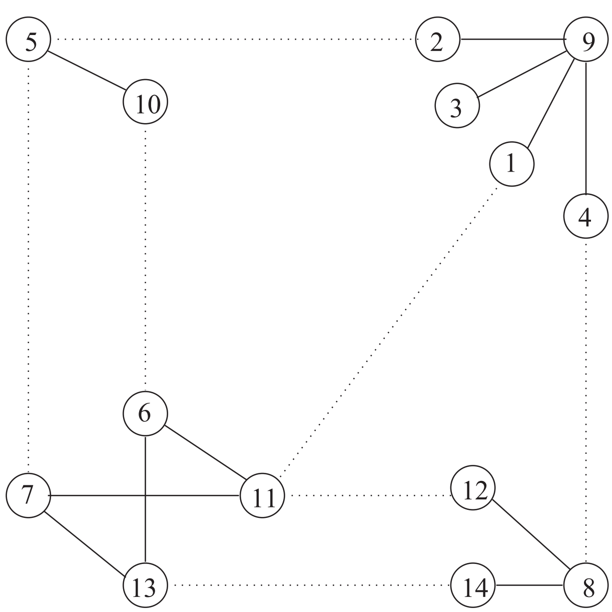

Fig. 2. The layout graph for the chip shown in Fig. 1.

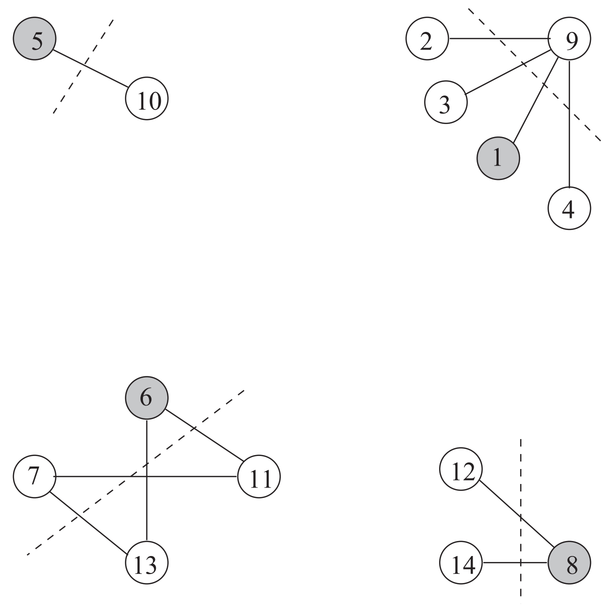

Fig. 3. Bipartite connected components of the graph in Fig. 2.

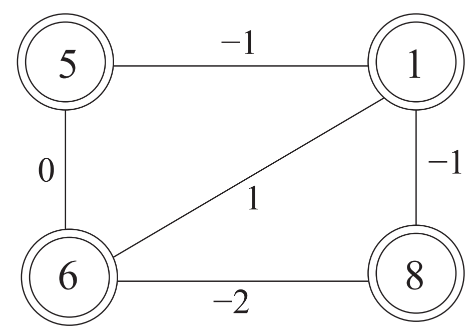

Fig. 4. The reduced graph of the graph in Fig. 2.

G contains a continuation edge between the two components. For each edge, weight is defined as the difference of the number of vias when the two nodes are assigned to the same layer and the number of vias when the two nodes are assigned to different layers. If we define weights like this way, the problem of minimizing the number of vias in the original graph G is equivalent to the max-cut problem in the reduced graph H by some mathematical manipulations. (For more details, refer to other papers [1,2]). The reduced graph H for our example is shown in Fig. 4. The maximum weight cut in H is {1, 5} and {6, 8} and it implies that the partition {1, 2, 3, 4, 5, 11, 12, 13, 14} and {6, 7, 8, 9, 10} of G is an optimal assignment of segments to two layers. This solution means that vias are placed between the critical segments 5 and 7 and between 4 and 8, because the critical segments 5 and 7 are assigned to different layers and there is a continuation edge between them, and the same goes for the critical segments 4 and 8.

There is another famous application of the max-cut problem in statistical physics [1]. It is concerned with the exact determination of the minimal energy configuration of the Ising spin glass under a continuously varying exterior magnetic field, or with no field. Poljak and Tuza [3] provided a comprehensive survey of the max-cut problem and its applications.

Every graph has a finite number of potential cuts, so it is possible to find the cut with the minimum or maximum weight by an exhaustive search. However, this is not a feasible approach for the large graphs that arise in practical applications, since the number of possible cuts in a graph grows exponentially with the number of vertices. If the requirement to balance the sizes of the subsets is dropped, then the min-cut problem can be solved in polynomial time using the maxflow-mincut algorithm [4]; but there is no equivalent approach for the max-cut problem, and exhaustive enumeration remains the only way of obtaining an optimal solution. Indeed, the max-cut problem was one of Karp's original NP-complete problems [5], and has been shown to be NP-complete even if the graph is unweighted [6].

It is therefore necessary to turn to heuristics, which become more rigorous if we seek a ρ-approximation algorithm, which can be proven to find a solution with a value that is at least some fraction ρ of the optimal value, in polynomial time. Sahni and Gonzales [7] presented