Applied Soft Computing 13 (2013) 2812–2822

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

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

Characterization of graph properties for improved Pareto fronts using heuristics and EA for bi-objective graph coloring problem

Soma Saha, Rajeev Kumar*, Gyan Baboo

Department of Computer Science & Engineering, Indian Institute of Technology, Kharagpur, WB 721302, India

ARTICLE INFO

Article history:
Received 18 January 2012
Received in revised form 26 June 2012
Accepted 27 June 2012
Available online 7 July 2012

Keywords:
Evolutionary algorithms
Multi-objective optimization
Hybridization
Graph coloring problem
Graph density
Pareto front
Rank histogram
Multi-objective evolutionary algorithm

ABSTRACT

Bi-objective graph coloring problem (BOGCP) is a generalized version in which the number of colors used to color the vertices of a graph and the corresponding penalty which incurs due to coloring the end-points of an edge with the same color are simultaneously minimized. In this paper, we have analyzed the graph density, the interconnection between high degree nodes of a graph, the rank exponent of the standard benchmark input graph instances and observed that the characterization of graph instances affects on the behavioral quality of the solution sets generated by existing heuristics across the entire range of the obtained Pareto fronts. We have used multi-objective evolutionary algorithm (MOEA) to obtain improved quality solution sets with the problem specific knowledge as well as with the embedded heuristics knowledge. To establish this fact for BOGCP, hybridization approach is used to construct recombination operators and mutation operators and it is observed from empirical results that the embedded problem specific knowledge in evolutionary operators helps to improve the quality of solution sets across the entire Pareto front; the nature of problem specific knowledge differentiates the quality of solution sets.

1. Introduction

Graph coloring problem (a.k.a. GCP) is a well-studied single-objective combinatorial optimization problem where the aim is to minimize the number of colors which is used to color the vertices of a given graph G without allowing the same color to the adjacent vertices. Approximating the chromatic number of a given graph G within range $n^{1-\epsilon}$ for any ϵ>0, where n is the cardinality of vertex set and deciding whether G is k ($k \geq 3$) colorable or not are two well-studied variant of single-objective GCP which belong to NP-hard and NP-complete class, respectively [1–3]. In this work, we have considered a bi-objective variant of GCP where the number of colors which is used to color the vertices of a given graph G and the penalty that incurs due to coloring adjacent vertices with the same color, are minimized.

The application areas of single objective graph coloring are timetable scheduling, examination scheduling, register allocation, printed circuit testing, electronic bandwidth allocation, microcode optimization, channel routing, the design of flexible manufacturing systems and others. In reality, it may not always be possible to allow the chromatic number of colors to solve the optimization problems. If the number of allowed color is smaller than the chromatic number, it is obvious that a penalty will occur and the goal of the single objective graph coloring may not be solved. With this practical point of view to find the coloring with minimum penalty if the number of allowed color is smaller than the chromatic number and the solution is acceptable with penalty, we aim to work on this bi-objective version of graph coloring problem.

Evolving heuristic algorithms that give approximate or sub-optimal solutions to the considered problem is a widely used method to solve NP-complete optimization problems [4]. Hence, numerous heuristics exist for single-objective GCP. Transformations can be used by heuristic algorithms to give approximate solutions to other NP-complete optimization problems [4]. Moreover, it is easier and less time-consuming to implement and develop than constructing a new heuristic algorithm from scratch. Thus, we have considered and adapted a few single-objective GCP heuristics such as Largest Degree Ordering (LDO) [5], DSatur/Saturated Degree Ordering (SDO) [6], Smallest Last Ordering (SLO) [7], Iterated Greedy [8], and Incidence Degree Ordering (IDO) [9] into the considered bi-objective variant of GCP. Combining heuristics to avoid the minor weaknesses of individual heuristics is a well-known approach to solve optimization problems. Al-Omari and Sabri [9] suggested two combined heuristics where individual LDO and SDO are modified and combined with IDO and LDO, respectively to produce better solution than individual heuristics for single-objective GCP. Well-combined single-objective heuristics for GCP

* Corresponding author.
E-mail addresses: rkumar@cse.iitkgp.ernet.in, rajeevkumar.cse@gmail.com (R. Kumar).

1568-4946/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2012.06.021

are adapted for bi-objective GCP and penalty adjusting heuristics (PAHs) [10] are considered in this work.

Competing goals and highly complex large search spaces of multi-objective optimization problems boost the need of multiple compromised solution set, known as Pareto-optimal set, instead of single optimal solution for a single objective/goal; hence, evolutionary approaches overpower the rest of the approaches. The considered variant of bi-objective graph coloring is a combinatorial explosive NP-hard problem. Studies have shown that such problems are difficult to solve using EA alone [11–13]. Multi-objective problems are much more challenging due to diversity of the converged solutions resulting the Pareto front. We have seen that MOEA alone takes enormous amount of time and even that the obtained solutions may not have good diverse and converged solutions. Thus, we have considered hybridization which is known as embedding problem specific knowledge and/or combining nature inspired search techniques.

Problem specific knowledge is embedded in Penalty based Color Partitioning Crossover (PCPX) and Degree Based Crossover (DBX) [10] operators and it has shown that Pareto Converging Genetic Algorithm (PCGA), a steady-state multi-objective evolutionary algorithm (MOEA), produces superior solution-set across the entire range of Pareto front in comparison with considered heuristics [10]. Han at el. proposed a bi-objective evolutionary algorithm (BEA) [14] for a bi-objective variant of GCP to produce optimal penalty for chromatic number. A few work have done using Multi-Objective Genetic Programming (MOGP) to evolve the hyper-heuristics for the considered bi-objective GCP [15] that produce the comparable solution-sets with the combined DSatur—LDO heuristic.

Characterization of the problem instances is needed to choose a particular algorithm for the individual problem instances with the prediction of performance in terms of the quality of solution, computational time, etc. It is also necessary to understand the behavior of an algorithm in advance level. Choosing a particular method *i.e.* either descriptive or empirical to characterize the problem at instance level is practically difficult. In this work, we have considered the graph density parameter and adapted a few static methods to characterize graph instances. Depending on the characterization of graphs at instance level, a particular solution method can be chosen *a priori* with the prediction of comparative solution quality. In this work, we have applied hybridization with EA on a bi-objective variant of GCP and analyzed whether the obtained Pareto front is comparable with the obtained Pareto fronts generated by the adaptation of a few well-known heuristics and by MOEA with a few crossover operators such as Penalty based Color Partitioning Crossover (PCPX) [10], Degree Based Crossover (DBX) [10] for the bi-objective variant of GCP. We have analyzed the nature of solution-sets across the complete range of Pareto fronts over each solution method and explored the change in behavior depending on the type of input graph instances.

Section 2 contains the problem formulation and description which includes the adapted heuristics in our work. The characterization of graph instances to analyze the nature of heuristics is described in Section 3. Section 4 describes the hybridization technique of multi-objective evolutionary algorithm which generates the improved quality of solution sets for the bi-objective graph coloring problem. Next, Section 5 contains the empirical results and comparative analysis of heuristics and MOEA solution sets. We draw conclusions in Section 6.

2. Problem formulation and description

Graph coloring with rejections is one of the standard bi-objective variants of graph coloring problem [16,17]. The problem is to select a subset of vertices V' over a set of vertices V for a given graph $G=(V, E)$, where each vertex v is assigned with a rejection cost and to find a proper coloring to the subgraph of G over V'. The objective of this problem is to minimize the total number of colors used to color V' and total rejection cost of all other vertices.

In this work, we have considered a variant of bi-objective GCP in which if adjacent vertices are colored with the same colors, a penalty is incurred. The problem is to minimize both the number of colors used to color the set of vertices V of a given graph $G=(V, E)$ and the penalty which is incurred due to coloring adjacent vertices with same color. We have used the phrase *bi-objective GCP* (BOGCP) in the rest of the sections for the considered variant of bi-objective GCP.

The considered heuristic frameworks are listed below.

1 **Smallest Degree Last**: In Smallest Last Ordering or Smallest Degree Last (SDL) heuristic [7], initially each vertex is assigned a degree according to the number of edges incident on that vertex. Next, smallest degree vertices are removed from the initial graph; the same procedure is applied to the subgraph repeatedly until vertices with degree zero (or highest priority vertices) remain in final subgraph. A sequence is maintained and that information is stored during the removal of vertices. A *sequential coloring* [18] is a k coloring for a graph G where each vertex v_i of ordering $\{v_1, v_2, \ldots, v_n\}$ is colored with the color $c(v_i)$ where $c(v_i) = \min\{m|1 \leq m \neq c(v_j)$ for v_j adjacent to $v_i, j < i\}$ and $k = \max\{c(v_i)|1 \leq i \leq n\}$. Highest priority vertices are colored first and least priority vertices (or smallest degree vertices) are colored at the end using *sequential coloring*.

Coloring sequence of vertices in Largest Degree Ordering (LDO) depends on the highest number of neighbors of a vertex. The term LDO was mentioned in [5,9] and it differs with SDL by sequential coloring. The complexity of LDO heuristic to find the minimum number of color for GCP is $O(V^2)$. LDO was termed as Largest First (LF) algorithm in [19].

2 **DSatur–LDO**: The maximum degree vertex from the decreasing degree-base ordered vertex set V of graph G is colored with a randomly chosen initial color in DSatur heuristic. Next, remaining vertices are chosen to color depending on maximal saturation degree which is known as the number of different colors to which a vertex is adjacent (colored vertices). LDO is used to break the saturation degree tie between two vertex. The least possible (or lowest number) color is used to color the chosen vertex. Al-Omari and Sabri [9] have claimed that for single objective GCP problem, DSatur–LDO/SDO–LDO generates better result than individual comprising heuristics and also LDO–IDO combined heuristic.

3 **DSatur–IDO–LDO**: A modification of DSatur/SDO heuristic is known as the Incidence Degree Ordering (IDO) [9]; the number of adjacent colored vertices is known as the incidence degree of a vertex. In DSatur–IDO–LDO heuristic, DSatur heuristic generated saturation degree ties are resolved by IDO with first priority. Remaining unresolved ties are broken by LDO.

We have considered two penalty adjusting heuristics which were proposed for BOGCP [10]. Initially, vertex coloring is started using two colors and stopped when penalty zero is reached for a particular value of color. With the aim to reduce the penalty for a given particular coloring, the entire vertex set of the graph is scanned and reassigned with a color which minimizes the penalty for that vertex. The second penalty adjusting heuristic chooses a color for replacement with a directed way, whereas the first penalty adjusting heuristic chooses a color randomly.

It is really difficult to choose a particular heuristic for a particular instance of the problem where the comparative nature of the heuristics performance depends on the type of the input instances. Moreover, characterization of the problem instances is itself a