Tabu Search for the Graph Coloring Problem

Enrico Malaguti, Paolo Toth

Dipartimento di Elettronica, Informatica e Sistemistica, University of Bologna
Viale Risorgimento, 2 - 40136 - Bologna (Italy)
e-mails: enrico.malaguti@unibo.it, paolo.toth@unibo.it

1 Tabu Search for the Graph Coloring Problem

Given an undirected graph $G = (V, E)$, the Graph Coloring Problem (GCP) requires to assign a color to each vertex in such a way that colors on adjacent vertices are different and the number of colors used is minimized. A subset of $V$ is called a stable set if no two adjacent vertices belong to it. A $k$ coloring of $G$ is a coloring which uses $k$ colors, and corresponds to a partition of $V$ into $k$ stable sets.

The Graph Coloring problem is in the original list of NP-hard problems (see Garey and Johnson [8]), and has received a large attention in the literature, not only for its real world applications in many engineering fields, including, among many others, scheduling [10], timetabling [4], register allocation [3], train platforming [2], frequency assignment [7] and communication networks [14], but also for its theoretical aspects and for its difficulty from the computational point of view.

Recently, Malaguti et al. [11] proposed an evolutionary algorithm combining a Tabu Search procedure, based on the Impasse Class Neighborhood (defined by Morgenstern in [13]), and a crossover operator.

The Impasse Class Neighborhood is a structure used to improve a partial $k$ coloring to a complete coloring of the same value, thus, a method which works with a fixed number of colors $k$ and partial colorings (not all vertices are colored). A solution $S$ is a partition of $V$ in $k + 1$ color classes $\{V_1, ..., V_k, V_{k+1}\}$ in which all classes, but possibly the last one, are stable sets. This means that the first $k$ classes constitute a partial feasible $k$ coloring, while all vertices that do not fit in the first $k$ classes are in the last one. Making this last class empty gives a complete feasible $k$ coloring. To move from a solution $S$ to a new solution $S'$ in the neighborhood, one can choose an uncolored vertex $v \in V_{k+1}$, assign $v$ to a different color class, say $h$, and move to class $k + 1$ all vertices $v'$ in class $h$ that are adjacent to $v$. This ensures that color class $h$ remains feasible.

Malaguti et al.[11] exploited this neighborhood in a Tabu search algorithm. The uncolored vertex $v$ is randomly chosen, while the class $h$ is chosen by comparing different target classes by means of an evaluating function $f(S)$ which minimizes the sum of the degrees of the uncolored vertices. This choice forces vertices having small degree, which are easier to color, to enter class $k + 1$. In order to avoid cycling, a tabu rule prevents an uncolored vertex $v$ from taking a color $h$ it recently took at a previous iteration.

The algorithm by Malaguti et al., integrated with a post-optimization procedure, can find 30 times the best known solution for a set of 34 hard graph coloring instances from the DIMACS benchmark, and in two cases it is the only algorithm in the literature able to
compute the best solution. The computing times of the algorithm range from few seconds for small graphs to approximately three hours for very large graphs (1000 vertices).

Another effective Tabu Search algorithm based on the impasse class neighborhood in proposed in Blöchliger and Zufferey [1], while a completely different neighborhood is considered in Hertz and de Werra [9], in which a solution is represented by a complete $k$ coloring of the graph (all vertices are colored), where some edges are conflicting, i.e., both endpoints share the same color.

The reader is referred to the recent survey by Malaguti and Toth [12] for a detailed review of algorithms and computational results for the graph coloring problem.

References


