Engineering Applications of Artificial Intelligence 37 (2015) 20–33

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

# General swap-based multiple neighborhood tabu search for the maximum independent set problem

Yan Jin, Jin-Kao Hao *

*LERIA, Université d'Angers, 2 Boulevard Lavoisier, 49045 Angers, France*

A R T I C L E I N F O

*Article history:*
Received 11 April 2014
Received in revised form
2 July 2014
Accepted 12 August 2014
Available online 15 September 2014

*Keywords:*
Maximum independent set
Maximum clique
Multiple neighborhoods
Local search
Tabu search

A B S T R A C T

Given a graph $G=(V,E)$, the Maximum Independent Set problem (MIS) aims to determine a subset $S \subseteq V$ of maximum cardinality such that no two vertices of $S$ are adjacent. This paper presents a general Swap-Based Tabu Search (SBTS) for solving the MIS. SBTS integrates distinguished features including a general and unified $(k,1)$-swap operator, four constrained neighborhoods and specific rules for neighborhood exploration. Extensive evaluations on two popular benchmarks (DIMACS and BHOSLIB) of 120 instances show that SBTS attains the best-known results for *all* the instances. To our knowledge, such a performance was not reported in the literature for a single heuristic. The best-known results on 11 additional instances from code theory are also attained.



## 1. Introduction

Given a simple undirected graph $G=(V,E)$ with vertex set $V=\{v_1,\ldots,v_n\}$ and edge set $E \subset V \times V$. An independent set $S$ is a subset of $V$ such that no two vertices are adjacent, i.e., $\forall v_i, v_j \in S, \{v_i, v_j\} \notin E$. An independent set is said maximum if it has the largest cardinality among all the independent sets of $G$. The maximum independent set problem (MIS) is to determine a maximum independent set of an arbitrary graph. As one of Karp's 21 NP-complete problems (Karp, 1972), MIS is among the most popular problems in combinatorial optimization (Garey and Johnson, 1979; Johnson and Trick, 1996).

In graph theory, there are two tightly related problems: the maximum clique problem (MC) and minimum vertex cover problem (MVC). A clique $C$ of $G$ is a subset of $V$ such that all vertices in $C$ are pairwise adjacent, i.e., $\forall v_i, v_j \in C, \{v_i, v_j\} \in E$. MC is to find a clique $C$ of maximum cardinality. A vertex cover $VC$ of $G$ is a subset of $V$ such that each edge of $E$ is incident to at least one vertex of $VC$, i.e., $\forall \{v_i, v_j\} \in E, v_i \in VC \vee v_j \in VC$. MVC is to determine a vertex cover of minimum cardinality.

Let $\overline{G}=(V,\overline{E})$ be the complementary graph of $G=(V,E)$ such that $\overline{E} \subset V \times V$ and $\forall v_i, v_j \in V, \{v_i, v_j\} \in \overline{E}$ if and only if $\{v_i, v_j\} \notin E$. Then given a subset $S$ of $V$, the following three statements are equivalent (Wu and Hao, 2014): $S$ is an independent set in $G$, $V \setminus S$ is a vertex cover in $G$ and $S$ is a clique in $\overline{G}$. As a consequence, MIS, MC and MVC are three equivalent problems such that any algorithm designed for one of these problems can be directly applied to solve the other two problems. These problems are relevant to a wide variety of applications such as code theory, information retrieval, signal transmission, classification theory, experimental design and many more others (Bomze et al., 1999; Johnson and Trick, 1996; Wu and Hao, 2014). In this work, we focus on studying the MIS problem.

During the past decades, a large number of solution procedures for solving MIS, MC and MVC have been reported in the literature. Among them are several exact algorithms based on the general branch-and-bound framework (Carraghan and Pardalos, 1990; Li and Quan, 2010; Östergård, 2002; San Segundo et al., 2011; Tomita and Kameda, 2007). These exact methods are applicable to problem instances of limited sizes. For larger cases, various heuristics have been proposed to obtain near-optimal solutions. The most representative heuristics include tabu search (Battiti and Protasi, 2001; Friden et al., 1989; Wu et al., 2012; Wu and Hao, 2013), stochastic local search (Andrade et al., 2012; Grosso et al., 2008; Katayama et al., 2005; Pullan, 2006, 2008), parallel hyper-heuristics mixing several low-level heuristics (Pullan et al., 2011), simulated annealing (Geng et al., 2007), variable neighborhood search (Hansen et al., 2004), breakout local search (Benlic and Hao, 2013), local search with edge weighting (Cai et al., 2013; Richter et al., 2007) and evolutionary algorithms (Brunato and Battiti, 2011; Zhang et al., 2005).). According to the reported results on benchmark instances, in particular those of the well-known Second DIMACS Implementation Challenge on Cliques, Coloring, and Satisfiability (Johnson and Trick, 1996), it seems that ILS and GLP

* Corresponding author.
*E-mail addresses:* jin@info.univ-angers.fr (Y. Jin), hao@info.univ-angers.fr (J.-K. Hao).

http://dx.doi.org/10.1016/j.engappai.2014.08.007

(Andrade et al., 2012), BLS (Benlic and Hao, 2013), NuMVC (Cai et al., 2013), PLS (Pullan, 2006, 2008), CLS (Pullan et al., 2011), COVER (Richter et al., 2007), MN/TS (Wu et al., 2012) and AMTS (Wu and Hao, 2013) are among the top performing heuristics in the literature. Nevertheless, due to the large variety of structures of these instances (they are random graphs or transformed from different real problems), no single approach can attain the best-known results for all the DIMACS instances.

In this work, we introduce a general Swap-Based Tabu Search (SBTS) heuristic for the maximum independent set problem. SBTS inspects the search space by a dynamic alternation between intensification and diversification steps (Glover and Laguna, 1997; Lourenço et al., 2003; Schrimpf et al., 2000). The search process is driven by a general and unified ($k$,1)-swap ($k \geq 0$) operator combined with specific rules to explore four constrained neighborhoods. Given an independent set $S$, ($k$,1)-swap exchanges one vertex (which is strategically selected) in $V \setminus S$ against its $k$ adjacent vertices in $S$. For the purpose of intensification, SBTS uses (0,1)-swap to improve the solution and (1,1)-swap to make side-walks with the help of specific selection rules. To overcome local optima, SBTS adopts an adaptive perturbation strategy which applies either a (2,1)-swap for a weak perturbation or a (k,1)-swap ($k>2$) for a strong perturbation. A tabu mechanism is also employed to prevent the search from short-term cycles. Compared with existing local search algorithms, SBTS distinguishes itself by its unified $(k,1)-$swap operator, its specific neighborhoods and its dedicated rules for neighborhood exploration.

The proposed SBTS algorithm attains the best-known results for all 120 instances of the well-known DIMACS and BHOSLIB benchmarks with very different structures and topologies. This is the first time a single heuristic reaches such a performance. The best-known results are also attained on an additional set of 11 real instances from code theory.

The rest of the paper is structured as follows. Section 2 describes the SBTS approach. Section 3 shows computational results and comparisons with the state-of-the-art algorithms in the literature. Before concluding, Section 4 investigates and analyzes some important issue of the proposed algorithm.

## 2. A swap-based tabu search for MIS

Our Swap-Based Tabu Search (SBTS) algorithm for MIS follows the iterated local search framework (Lourenço et al., 2003) and shares similarities with other methods like variable neighborhood search (Hansen et al., 2004) and the ruin-and-recreate search (Schrimpf et al., 2000). However, as we explain in this section, SBTS possesses some particular features like four constrained neighborhoods and the specific rules for an effective exploration of these neighborhoods.

### 2.1. General procedure

The general SBTS procedure is summarized in Algorithm 1. SBTS uses a fast randomized construction procedure (Section 2.3) to obtain a first feasible independent set $S$ (i.e., no two vertices of $S$ are adjacent, $S$ is also called *a feasible solution* or simply *a solution* in the paper). From this initial solution, SBTS tries to find improved solutions (i.e., larger independent sets) by a series of intensification and diversification steps (Sections 2.7 and 2.8). Both intensification and diversification steps are based on the general $(k,1)-$swap operator (Section 2.5).

Specifically, each intensification step makes a $(k,1)-$swap move ($k=0,1$) to increase the cardinality of the independent set or search new solutions while keeping the cardinality unchanged. Inversely, a diversification step applies a $(k,1)-$swap move ($k \geq 2$) to decrease temporarily the quality of the current solution (the current solution loses $k-1$ vertices). Whenever there exist intensification moves, they are always preferred over diversification moves. Diversification moves are only applied to escape from a local optimum (i.e., when no eligible $(k,1)-$swap move ($k=0,1$) is available). As we explain in Sections 2.7 and 2.8, both intensification and diversification are subject to dedicated rules which govern the way $(k,1)-$swap moves are executed.

SBTS uses a global variable $S_*$ to record the best solution ever discovered during the search and a tabu list to prevent short-term cycles (see Section 2.6). The algorithm stops when a fixed number of iterations are realized.

**Algorithm 1.** General procedure of the SBTS algorithm for MIS.

```
Input: A graph G, Iters_max (maximum allowed iterations
       per run)
Output: The largest independent set S* found.
S ← Initialization()   /* Generate a feasible independent set
    S, Sect. 2.3 */
S* ← S   /* S* records the largest independent set found so
    far */
f* ← f(S)   /* f* records the cardinality of S* */
Initialize tabu_list   /* Initialize the tabu list, Section 2.6 */
for iters ← 1 to Iters_max do
  if there exists an eligible intensification move then
    S ← IntensificationStep(S)   /* Apply (k,1)−swap
    (k ≤ 1) to improve solution S, Section 2.7 */
    if f(S) > f* then
      S* ← S, f* ← f(S)
    end if
  else
    S ← DiversificationStep(S)   /* Apply (k,1)−swap
  (k > 1) to perturb solution S, Section 2.8*/
  end if
  Update tabu_list /* Section 2.6 */
end for
return S*
```

### 2.2. Search space and evaluation function

Before presenting the components of the SBTS algorithm, we define first the search space $\Omega$ explored by the algorithm as well as its evaluation function $f$ to measure the quality of a candidate solution.

For a given graph $G=(V,E)$, the search space $\Omega$ explored by SBTS is the set of all the independent sets of $G$, i.e., $\Omega = \{S \subseteq V : v_i, v_j \in S, \{v_i, v_j\} \notin E\}$. For any feasible solution $S \in \Omega$, its quality is directly assessed by the cardinality of $S$, i.e., $f(S)=|S|$. Given two independent sets $S$ and $S'$, $S$ is better than $S'$ if and only if $f(S) > f(S')$.

### 2.3. Initial solution

The initial solution used by the SBTS algorithm is generated by the following sequential randomized heuristic ($V$ is the vertex set of graph $G$).

1. Set $S$ to empty.
2. Select randomly a vertex $u \in V$ and add $u$ into $S$.
3. Remove from $V$ vertex $u$ and all its adjacent vertices $v \in V$ ($\{u,v\} \in E$).
4. Repeat steps (2)–(3) until $V$ becomes empty and return $S$.

It is easy to observe that the resulting solution $S$ is a feasible (and maximal) independent set. Due to the random choices at