SCIENCE CHINA
Information Sciences

May 2021, Vol. 64 150105:1–150105:2
https://doi.org/10.1007/s11432-020-3069-4

• LETTER •

Special Focus on Constraints and Optimization in Artificial Intelligence

Solving diversified top-k weight clique search problem

Junping ZHOU[1], Chumin LI[2], Yupeng ZHOU[1], Mingyang LI[1],
Lili LIANG[1] & Jianan WANG[1]*

[1]*School of Information Science and Technology, Northeast Normal University, Changchun* 130117*, China;*
[2]*Modélisation, Information & Systèmes, Université de Picardie Jules Verne, Amiens* 80000*, France*

Received 18 May 2020/Accepted 4 August 2020/Published online 29 March 2021

Citation Zhou J P, Li C M, Zhou Y P, et al. Solving diversified top-k weight clique search problem. Sci China Inf Sci, 2021, 64(5): 150105, https://doi.org/10.1007/s11432-020-3069-4

Dear editor,

Diversified top-k weight clique search (DTKWCS) is a problem that computes k cliques to maximize the sum of weights of all vertices contained in the cliques; that is, $\Sigma_{v\in\{c_1\cup c_2\cup\cdots\cup c_k\}}w(v)$ is maximized by giving a weighted graph G and an integer k, where c_i is one of the k cliques, and $w(v)$ is the weight of the vertex v in G. This problem is NP-hard. It can be applied in spectrum sharing problem, advertisement putting problem, gene expression and motif discovery, influential community search, sensor place problem, and anomaly detection in complex networks [1–4]. In solving DTKWCS in an unweighted graph, a trivial direct approach based on all cliques enumeration is used [5]; however, the approach is time-consuming and not suitable for solving large graphs. Another direct-solving approach is proposed that can give approximate solutions [6]; however, the approach is not competitive in solving dense graphs and cannot guarantee the optimality of its solutions. Therefore, it is worth exploring a generic approach to solving DTKWCS.

In this study, we provide a generic approach for solving DTKWCS, which is done by encoding the DTKWCS into the weighted partial MaxSAT (WPMS) problem and then solving WPMS with state-of-the-art solvers. It has been proven that solving NP problems, including academic and industrial problems, by encoding as SAT or WPMS is an efficient strategy [7]. To perform the encoding of DTKWCS to WPMS, we present two encodings strategies: direct encoding (DE) and independent set partition based encoding (ISPE). As shown in the supporting information, the experimental results show that the two encoding strategies are competitive.

Preliminaries. $G=\langle V,E,w\rangle$ is an undirected weighted graph, where V and E are sets of vertices and edges, respectively, and w is a weight function that assigns a nonnegative integer, called weight, to each vertex v. A clique c_i of a graph G is a subset of vertices in G such that every two distinct vertices in the subset are adjacent. A literal is either a Boolean variable (variable for brevity in the rest of study) x or its negation $\neg x$. A clause is a disjunction of literals, which is satisfied if and only if at least one literal in it taking the value true. A weighted clause is a pair (c,w), where c is a clause, and w is the weight of the clause. A weighted clause is hard if its weight is infinite; otherwise, the clause is soft. A WPMS formula F in CNF is a conjunction of hard and soft clauses. The purpose of the WPMS problem is to find a truth assignment for F by satisfying all hard clauses and then maximizing the sum of weights of all satisfied soft clauses.

Direct encoding. The basic idea of the DE is derived from the following observations. First, because the DTKWCS problem requires to find k cliques, we encode each vertex into k variables; that is, the vertex v_i is expanded into the variables $x_{i1},x_{i2},\ldots,x_{ik}$. Thus, the variable $x_{ij}=$ true if and only if the vertex v_i is in the jth clique. Second, the DTKWCS and WPMS problems are both used to compute a solution to maximize the sum of weights of vertices (or soft clauses). Then, DE encoding creates hard clauses that could guarantee every feasible solution of a WPMS instance to form k cliques. Finally, the DE encoding employs a direct way to encode soft clauses; that is, each vertex v_i defines a soft clause, which is satisfied if and only if v_i is in at least one of the k cliques.

Formally, given a graph $G=\langle V,E,w\rangle$ and an integer k, we define the DE encoding as follows.

(1) For each $v_i\in V$, create k variables $x_{i1},x_{i2},\ldots,x_{ik}$.

(2) For any two unconnected vertices v_i and v_j in V (i.e., $\langle v_i,v_j\rangle\notin E$), create k hard clauses: $(\neg x_{i1}\vee\neg x_{j1},\infty)$, $(\neg x_{i2}\vee\neg x_{j2},\infty)$, … , $(\neg x_{ik}\vee\neg x_{jk},\infty)$.

(3) For each vertex $v_i\in V$, create a soft clause $(x_{i1}\vee x_{i2}\vee\cdots\vee x_{ik},w(v_i))$.

We denote the resulting WPMS formula by ϕ. The DE encoding has the following properties.

- Any feasible solution of ϕ, that is, any truth assignment satisfying all hard clauses of ϕ, gives k cliques. To see this, let us partition the variables assigned with the value true into the k subsets: $\{x_{i_{11}1},x_{i_{12}1},x_{i_{13}1},\ldots\}$, $\{x_{i_{21}2},x_{i_{22}2},x_{i_{23}2},\ldots\},\ldots,\{x_{i_{k1}k},x_{i_{k2}k},x_{i_{k3}k},\ldots\}$. Any two vertices corresponding to two variables in a subset, saying $v_{i_{j1}}$ and $v_{i_{j2}}$, must be adjacent; otherwise, a hard clause

[*] Corresponding author (email: wangjn@nenu.edu.cn)

$\neg x_{i_{j1}j} \vee \neg x_{i_{j2}j}$ was created because $v_{i_{j1}}$ and $v_{i_{j2}}$ were not adjacent, which would be falsified.

- Any k cliques give a truth assignment satisfying all hard clauses of ϕ: x_{ij} = true if and only if the vertex v_i is in the jth clique.
- Any optimal solution of ϕ gives k cliques covering the vertices with the maximum sum of weights. To see this, note that each soft clause corresponds to a unique vertex, and it is satisfied if and only if the corresponding vertex is covered. Thus, the maximum sum of weights of the satisfied soft clauses is equal to the maximum sum of weights of the covered vertices.
- In the worst case, the time complexities of DE encoding is $O(|V|^2)$, where the complexity of generating variables and generating clauses are $O(|V|)$ and $O(|V|^2)$, respectively.

Independent set partition based encoding. The ISPE includes two conversion processes: reducing a DTKWCS instance into a new version of partial MaxSAT, named literal WPMS (LWPMS), and subsequent transforming LWPMS into WPMS. Before the introduction of the ISPE encoding, some related definitions are given. LWPMS is a conjunction of hard clauses and literal-weighted soft clauses. The literal-weighted soft clause is composed of weighted literals denoted by (l, w), where l is a literal, and w is the weight of the literal. Given a graph G, we note that an independent set is a set of disconnected vertices.

Next, we present the first part of ISPE encoding from DTKWCS into LWPMS by given a graph $G = \langle V, E, w\rangle$ and an integer k as follows.

(1) For each $v_i \in V$, create k variables $x_{i1}, x_{i2}, \ldots, x_{ik}$.

(2) For any two unconnected vertices v_i and v_j in V (i.e., $\langle v_i, v_j\rangle \notin E$), create k hard clauses: $(\neg x_{i1} \vee \neg x_{j1}, \infty)$, $(\neg x_{i2} \vee \neg x_{j2}, \infty)$, ... , $(\neg x_{ik} \vee \neg x_{jk}, \infty)$.

(3) For each $v_i \in V$, create $\binom{k}{2}$ hard clauses. Specifically, for any two variables x_{ir} and x_{ij} $(r \neq j, 1 \leqslant r, j \leqslant k)$ generated by v_i, create a hard clause $(\neg x_{ir} \vee \neg x_{ij}, \infty)$.

(4) Partition the graph G into several disjoint independent sets, and ensure that the vertices in the disjoint independent sets constitute V. Then for each independent set $\{v_i, v_j, \ldots, v_r\}$, create k literal-weighted soft clauses $(x_{is}, w(v_i)) \vee (x_{js}, w(v_j)) \vee \cdots \vee (x_{rs}, w(v_r))$ $(s = 1, 2, \ldots, k)$.

The following is the intuition behind the first part of ISPE encoding. The hard clauses generated by the disconnected vertices guarantee that the vertices build up k cliques. To ensure that the k cliques are not duplicated, we generate $\binom{k}{2}$ hard clauses for each vertex. Furthermore, literal-weighted soft clauses guarantee that the sum of weights of covered vertices is maximum. In view that no existing solvers can solve the LWPMS, we manage to encode LWPMS into WPMS in the second part. By comparing LWPMS and WPMS, we understand that the difference between both is the type of soft clauses: literal weighted and clause weighted. Therefore, we need to transform the literal-weighted soft clauses into clause-weighted soft clauses (i.e., soft clauses). The method of the conversion is done by iteratively splitting the weighted literals. For each literal-weighted soft clause, we generate a set of soft clauses, as shown in Algorithm 1. After encoding all literal-weighted soft clauses into soft clauses, the LWPMS is reduced into WPMS. Similarly, $x_{ij} = 1$ if and only if the vertex v_i is in the jth clique. The optimal solution of the WPMS instance corresponds to the maximum total weights of the covered vertices. We analyse the time complexity of the ISPE encoding in the worst case as follows. The complexities in the first and second parts are $O(|V|^2)$ and $O(|V|)$, respectively. Thus, in the worst case, the time complexity of the ISPE encoding is $O(|V|^2)$.

Algorithm 1 To-Soft-Clause

```
Input: a literal-weighted soft clause $(x_{is}, w(v_i)) \vee (x_{js}, w(v_j)) \vee \cdots \vee (x_{rs}, w(v_r))$;
Output: a set of soft clauses $C$.
$W \leftarrow \{w(v_i), w(v_j), \ldots, w(v_r)\}$;
$L \leftarrow \{x_{is}, x_{js}, \ldots, x_{rs}\}$;
$\delta \leftarrow \min W$;
while $L \neq \emptyset$ do
    Add a soft clause $(c, \delta)$ to $C$, where $c$ is a disjunction of all literals in $L$;
    Delete the weights equivalent to $\delta$ from $W$ and the literals whose weight is equal to $\delta$ from $L$;
    $W$ is updated by each weight in $W$ minus $\delta$;
    $\delta \leftarrow \min W$;
end while
return $C$.
```

To test the DE and ISPE encodings, we perform experiments using the approximate DTKWCS solver EnumKOpt [6], the exact WPMS solver RC2-2018 [8], and the heuristic WPMS solver TT-Open-wbo-Inc [9]. These solvers are considered to be the best in their category. The experimental results in Appendix A show that DE and ISPE encodings are efficient and effective.

Conclusion. This study defines two encoding strategies for solving the DTKWCS problem into the WPMS problem. It can be noted that DE encoding is a direct way, whereas ISPE is based on independent set partition. The experimental results show that our encoding strategies are efficient and effective, which also remedy the lack of dedicated exact solvers for the DTKWCS problem.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61976050, 61403076, 61806050, 61806082) and Fundamental Research Funds for the Central Universities (Grant No. 2412019FZ050).

Supporting information Appendix A. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Zheng X Q, Liu T G, Yang Z N, et al. Large cliques in Arabidopsis gene coexpression network and motif discovery. J Plant Physiol, 2011, 168: 611–618

2 Conrad L, Aaron M, Fergal R, et al. Detecting highly overlapping community structure by greedy clique expansion. In: Proceedings the 4th SNA-KDD Workshop on Social Network Mining and Analysis, Washington, 2010. 112–119

3 Krause A, Guestrin C. Near-optimal observation selection using submodular functions. In: Proceedings AAAI Conference on Artificial Intelligence, Canada, 2007. 22–26

4 Berry N M, Ko T H, Moy T, et al. Emergent clique formation in terrorist recruitment. In: Proceedings the AAAI Workshop on Agent Organizations: Theory and Practice, Menlo Park, 2004

5 Feige U. A threshold of ln n for approximating set cover. J ACM, 1998, 45: 634–652

6 Yuan L, Qin L, Lin X M, et al. Diversified top-k clique search. VLDB J, 2016, 25: 171–196

7 Li C M, Manyà F, Quan Z, et al. Exact MinSAT solving. In: Proceedings the 13th International Conference on Theory and Applications of Satisfiability Testing, Edinburgh, 2010. 363–368

8 Alexey I, Morgado A, Marques-Silva J. RC2: an efficient MaxSAT solver. J Satisfiability Boolean Modeling Comput, 2019, 11: 53–64

9 Fahiem B, Matti J, Ruben M. MaxSAT Evaluation 2019: Solver and Benchmark Descriptions. Department of Computer Science Report Series B-2019-2. 2019