ALGORITHM FOR THE VERTEX PACKING PROBLEM

Yu. V. Voitishin and E. A. Sharkovskaya

UDC 519.854.2

An algorithm to find the largest internally stable set of a graph is proposed. The algorithm is based on reduction to the problem of pairwise incomparable vertices with additional constraints.

Let $G = (V, E)$ be a finite undirected graph without loops and multiple edges; $V = \{v_i\}$ the vertex set of $G$, $E = \{e_j\}$ the set of edges of $G$ (edges are also denoted by $(v_i, v_j)$, $v_i, v_j \in V$). To each vertex $v_i \in V$ we associate some positive number $c(v_i)$, called the "weight" of the vertex $v_i$. The vertices $v_i$ and $v_j$ are called adjacent if $(v_i, v_j) \in E$; otherwise, $v_i$ and $v_j$ are nonadjacent vertices. The vertex packing problem involves finding a subset of nonadjacent vertices with a maximum sum of weights. This NP-complete problem is often called in the literature the maximum internally stable set problem.

The procedure $P_1$ transforms the graph $G$ by the transitive orientation algorithm proposed in [1]. The procedure $P_1$ orients the edges $(v_i, v_j) \in E$. Denote $R(v_i) = \{v_j \in V: (v_i, v_j) \in E \vee (v_j, v_i) \in E \vee (v_i, v_l) \in E\}$, where $[v_i, v_j]$ is an arc oriented from $v_i$ to $v_j$. Some vertices $v_l, l \in L$, are possibly split into two vertices each. In this case, we say that the vertex $v_l$ and the vertex added to the graph $G$ by the splitting of $v_l$ are split vertices. The procedure $P_1$ uses rules 1 and 2 of the algorithm of [1] in a modified form.

**Modified rule 1.** If $[v_i, v_k] \in E$, $(v_k, v_j) \in E$, and $v_k \notin R(v_i)$, then orient the edge $(v_k, v_j)$ as $[v_k, v_j]$ and do the procedure $P_2(v_k, v_j)$ described below.

**Modified rule 2.** If $(v_i, v_j) \in E$, $(v_i, v_k) \in E$, and $v_k \notin R(v_j)$, then orient the edge $(v_i, v_k)$ as $[v_i, v_k]$ and do the procedure $P_2(v_i, v_k)$.

**Procedure $P_1$**

**Step 1.** Choose the edge $(v_i, v_j)$ of the graph $G$ and orient it arbitrarily, as $[v_i, v_j]$ say. Execute the procedure $P_2(v_i, v_j)$. Using modified rules 1 and 2, orient if possible the edges of the graph $G$ adjacent to the arc $[v_i, v_j]$. Mark the arc $[v_i, v_j]$ as "inspected".

**Step 2.** Check if the graph $G$ contains an arc that has not been marked "inspected". If yes, go to step 3; if no, go to step 4.

**Step 3.** Assume that the arc $[v_i, v_j]$ is not marked "inspected". For each edge (arc) incident on $v_i$ or $v_j$ do the following (whenever possible), then mark the arc $[v_i, v_j]$ as inspected, and go to step 2.

**Case 1.** Let $(v_j, v_k)$ be the current edge.

A. If $v_k \notin R(v_i)$ and the edge $(v_j, v_k)$ is not oriented, then orient the edge $(v_j, v_k)$ as $[v_k, v_j]$ and do the procedure $P_2(v_k, v_j)$.

B. If $v_k \notin R(v_i)$ and the edge $(v_j, v_k)$ is already oriented as $[v_j, v_k]$, then the graph $G$ is not transitively orientable.

In this case, do the following. Check if $v_j$ is a split vertex. If no, split the vertex $v_j$ into two vertices $v^0_j$ and $v^1_j$; for the vertex $v^0_j$ duplicate all the edges and arcs incident on the vertex $v_j$, except the arc $[v_j, v_k]$; for the vertex $v^1_j$ duplicate all the edges and arcs incident on the vertex $v_j$, except the arc $[v_i, v_j]$. If yes, delete the arcs $[v_i, v^1_j]$ and $[v^0_j, v_k]$ (if they have not been deleted previously).

C. If $v_k \notin R(v_i)$ and the edge $(v_j, v_k)$ is already oriented as $[v_k, v_j]$, continue to inspect the next edge (arc).

**Case 2.** Let $(v_i, v_k)$ be the current edge.

A. If \( v_k \notin R(v_j) \) and the edge \((v_i, v_k)\) is not oriented, then orient the edge \((v_i, v_k)\) as \([v_i, v_k]\) and apply the procedure 

\[-\]

B. If \( v_k \notin R(v_j) \) and the edge \((v_i, v_k)\) is already oriented as \([v_k, v_i]\), then the graph \( G \) is not transitively orientable. In this case, do the following. Check if \( v_i \) is a split vertex. If no, split the vertex \( v_i \) into two vertices \( v_i^0 \) and \( v_i^1 \); for the vertex \( v_i^0 \) duplicate all the edges and arcs incident on the vertex \( v_i \), except the arc \([v_i, v_j]\); for the vertex \( v_i^1 \) duplicate all the edges and arcs incident on the vertex \( v_i \), except the arc \([v_k, v_i]\); go to step 2. If yes, delete the arcs \([v_k, v_i^1]\) and \([v_i^0, v_j]\) (if they have not been deleted previously); if this deletes the arc \([v_i, v_j]\) selected for inspection in step 3, then go to step 2.

C. If \( v_k \notin R(v_j) \) and the edge \((v_i, v_k)\) has already been oriented as \([v_i, v_j]\), then go to the next edge (arc).

**Step 4.** Check if all the edges of the graph \( G \) have been oriented. If yes, stop; if no, remove all the arcs from \( G \), producing the graph \( G' \); set \( G = G' \) and go to step 1.

**Procedure P2**

**Step 1.** Let

\[
\alpha = \begin{cases} 
1 & \text{if } v_j \text{ is a split vertex}, \\
0 & \text{otherwise}; 
\end{cases} \\
\beta = \begin{cases} 
1 & \text{if } v_j \text{ is a split vertex}, \\
0 & \text{otherwise}. 
\end{cases}
\]

If \( \alpha = 0 \) and \( \beta = 0 \), then stop.

**Step 2.** Three cases are possible:

a) \( \alpha = 1 \) and \( \beta = 0 \); then orient the edge \((v_i^*, v_j^\#)\) as \([v_i^*, v_j]\), where \( v_i^* \) is the "other half" of the vertex \( v_i \);

b) \( \alpha = 0 \) and \( \beta = 1 \); in this case, orient the edge \((v_i, v_j^\#)\) as \([v_i, v_j^\#]\);

c) \( \alpha = 1 \) and \( \beta = 1 \); here orient the edge \((v_i, v_j^\#)\) as \([v_i, v_j^\#]\) and the edge \((v_i^*, v_j^\#)\) as \([v_i^*, v_j^\#]\).

End.

Let \( \tilde{G} = [\tilde{V}, \tilde{E}] \) be the resulting directed graph (digraph), \( \tilde{V} = \{v_i\} \) the set of supervertices. A supervertex \( v_i \) is either a simple vertex, if the corresponding vertex in the graph \( G \) is not split, or the pair of vertices \( v_i^0 \) and \( v_i^1 \).

Two vertices are called comparable if there is a path in the digraph passing through these vertices. Two supervertices \( v_i \) and \( v_j \in \tilde{V} \) are called comparable if at least one of the component vertices of \( v_i \) is comparable with at least one of the component vertices of \( v_j \).

From the description of the procedures P1 and P2 it follows that the supervertices \( v_i \) and \( v_j \) of the digraph \( \tilde{G} \) are comparable if and only if the corresponding vertices of the graph \( G \) are adjacent.

To each supervertex of the digraph we associate a weight equal to the weight of the corresponding vertex of the graph \( G \). Then the vertex packing problem for the graph \( G \) is equivalent to the problem of the maximum set of pairwise incomparable