ROMAI J., **2**, 1(2006),
71–76

# INVERSE PROBLEMS OF TRANSITIVE CLOSURE

Adrian Deaconu
*Univ. "Transilvania" of Braşov, Faculty of Mathematics and Informatics*
Deaconu@info.unitbv.ro

**Abstract** A graph G' starting from a self-transitive closure graph G with as least as possible arcs so that the transitive closure of G' is G is found. Two algorithms are presented. The number of optimal solutions is computed.

**Keywords:** graphs, permutations

## 1. INVERSE PROBLEMS OF TRANSITIVE CLOSURE

Let G = (V, E) be an oriented graph.

**Definition 1.** The oriented graph G* = (V, E*) is called the *transitive closure* of the graph G, where $E^* = \{(x, y)|$ there is a directed path from x to y$\}$.

**Definition 2.** The oriented graph G = (V, E) is a *self-transitive closure* iff its transitive closure does not differ from G, i.e. G* = G.

Let us consider a self-transitive closure graph G = (V, E). The inverse problem of transitive closure (ITC) is to find $G^{-1} = (V, E^{-1})$, where $E^{-1}$ has as least arcs as possible from G and the transitive closure of $G^{-1}$ is G, i.e. $(G^{-1})^* = G$. Let us present other two definitions, which make the difference between the quality of the solutions of the inverse transitive closure problem.

**Definition 3.** $G^{-1} = (V, E^{-1})$ is called a *solution* for the inverse problem of transitive closure, if and only if $(G^{-1})^* = G$ and the elimination of any arc (x, y) from $E^{-1}$ leads to a graph $G' =$ (V, $E'$), where $E' = E^{-1} - \{(x, y)\}$, for which the transitive closure is not G, i.e. $((G')^* \neq G$.

**Definition 4.** $G^{-1} = (V, E^{-1})$ is called an *optimal solution* for the inverse problem of transitive closure, if and only if any graph $G' =$ (V, $E'$) so that $(G')^* =$ G has at least the number of arcs of $G^{-1}$, i.e. $|E'| \geq |E^{-1}|$.

Any optimal solution satisfies the inverse problem of transitive closure.

## 2. ALGORITHMS FOR THE ITC PROBLEM

The first idea that appears for solving the inverse problem is to find criteria for elimination of arcs from E so that the resulted graph $G' =$ (V, $E'$) ($E' =$

$E$ – {eliminated arcs}) still has graph G as its transitive closure. Here it is such a criterion:

**Theorem 1.** *If an arc (x, y) is eliminated from the graph G and in the resulted graph $G'$ = (V, $E'$) ($E' = E - \{(x, y)\}$) there is still a directed path from x to y, then $(G')^*$ = $G^*$.*

**Proof.** Let $(x, y) \in E$ be an arc so that in the graph $G' = (V, E' = E - \{(x, y)\})$ there is a directed path $P'$ from $x$ to $y$ (1). Suppose that $(G')^* \neq G^* \Rightarrow \exists (u, v) \in E^*$ and $(u, v) \notin (E')^*$ iff there is a directed path $P$ from $u$ to $v$ in G* and there is no directed path from $u$ to $v$ in $(G')^*$.

There are two situations:

1. if the arc $(x, y) \notin P$, then all arcs from P are in E and $E'$, so P is a directed path in $G'$ (contradiction);

2. if the arc $(x, y) \in P$, then $P = (u, ..., x, y, ..., v)$. Let $P''$ be the directed path P, where the arc (x, y) is replaced by the path $P'$ (see (1)), i.e. $P''$ = (u, ..., $P'$, ..., v) = (u, ..., x, ..., y, ..., v), where all arcs are in $E'$. So, there is a directed path $P''$ in $G'$ from $u$ to $v$ (contradiction).

In both cases we obtained contradiction, hence the assumption $(G')^* \neq G^*$ is false. So, $(G')^*$ = G* and the theorem is proved.

Starting with Theorem 1 and remarking that the arcs $(x, x), x \in V$ can be also eliminated from G, we can easily write an algorithm for finding a solution of the inverse problem of transitive closure.

```
1.   E⁻¹ := E \ {(x,x)|x∈V};
2.   for (x,y) ∈ E⁻¹ do
          if ∃ directed path from x to y in G'=(V,E'),
               where E'=E⁻¹ \ {(x,y)}
          then E⁻¹ := E⁻¹ \ {(x,y)};
          end if;
     end for;
```

*Fig. 1.* **Algorithm 1** (elimination of arcs).

**Theorem 2.** *The algorithm 1 finds a solution $G^{-1} = (V, E^{-1})$ of the inverse problem of transitive closure in a complexity of $O(m^2)$, where m is the number of directed arcs in G = (V, E), i.e. $m = |E|$.*

**Proof.** Obviously $G^{-1}$ is a solution, because every directed arc was tested for elimination, such that as a result of application of the algorithm 1 there is no other arc in $E^{-1}$ which can be eliminated any more. The test if there is a directed path from a node x to a node y can be done in a complexity of *O(m)*, using a search algorithm (BFS or DFS). There are m tests, so the complexity of the algorithm 1 is $O(m^2)$.

The initial graph G is a self-transitive closure graph and due to that, usually it has many arcs, it is a so-called dense graph. So, algorithm 1 is slow, often it runs in a complexity of $O(n^4)$, because in a dense graph $m$ is closed to

$n^2$. There is another problem with the algorithm 1. It finds a solution of the inverse problem, which is not necessarily optimal. Here is an example to illustrate this case.

Let us apply algorithm 1 to the graph in the fig. 2.

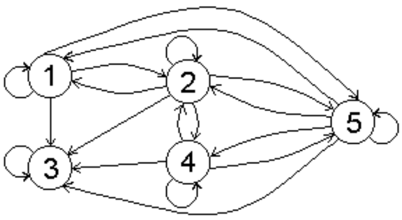


*Fig. 2.* An example.

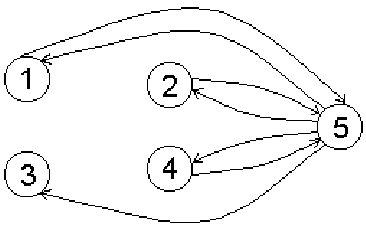


*Fig. 3.* Solution found by algorithm 1.

First the algorithm eliminates the arcs (1, 1), (2, 2), (3, 3), (4, 4), (5, 5). Suppose that in the second step the arcs (1, 2), (1, 3), (1, 4), (2, 1), (2, 3), (2, 4), (4, 1), (4, 2), (4, 3) are eliminated. Finally, it is obtained the graph $G^{-1}$ from the fig. 3. At the end of the algorithm, the graph $G^{-1}$ has 7 directed arcs. This is a solution of the inverse problem, but it is not optimal, because an optimal solution has only 5 directed (fig. 4).

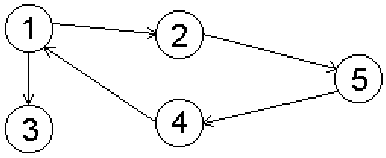


*Fig. 4.* Optimal solution.

Let us present a faster algorithm and prove that the found solution is optimal. The idea of this algorithm starts with the remark that the strongly connected components of any solution $G^{-1}$ of the inverse problem are the same with the strongly connected components of the initial graph G.

**Definition 5.** $K \subseteq V$ is a *strongly connected component* of the graph G = (V, E) if for each nodes $u$, $v \in K$ there is a directed path in G from $u$ to $v$.

**Definition 6.** The graph $G^c = (V^c, E^c)$ is called the *condensed graph* of G = (V, E) if it has as nodes all the strongly connected components of G and the directed arcs of $G^c$ are the connections between the strongly connected components of G, i.e. $V^c = \{K|K$ is a strongly connected component of $G\}$ and $E^c = \{(K_1, K_2)|K_1, K_2 \in V^c$ and $\exists u \in K_1, v \in K_2 : (u, v) \in E\}$.

Using as few directed arcs from E as possible for each strongly connected component so that it remains strongly connected and using the directed arcs of the condensed graph, then an optimal solution for the inverse problem of the transitive closure is obtained.

**Theorem 3.** *In a self-transitive closure graph G there is a directed path from x to y ($x, y \in V$), iff there is a directed path from x to y in a solution of the inverse problem denoted by* $G^{-1}$.

**Proof.** " <= " If a directed path P is from $G^{-1}$, as $E^{-1} \subseteq E$, it exists in G. " => " Let $P = (x = u_1, u_2, ..., u_r = y)$ be a directed path in G. For any directed arc $(u_i, u_{i+1})(i \in 1, 2, ?, r-1)$ from P, there is a directed path $P_i$ in $G^{-1}$, from $u_i$ to $u_{i+1}$. We replace every arc $(u_i, u_{i+1})$ with the path $P_i$ in P. It is obtained the path $P' = (Pu_1, Pu_2, ?, Pu_{r-1})$ from x to y.

**Theorem 4.** *$K \subseteq V$ is a strongly connected component of a self-transitive closure graph G = (V, E), iff K is a strongly connected component of any solution $G^{-1}$ of the ITC problem.*

**Proof.** Directly from Theorem 3.

**Theorem 5.** *If $K \subseteq V$ is a strongly connected component of a self-transitive closure graph G = (V, E), then the graph $G_K = (K, E_K)$ is complete, where $E_K = \{(x, y) \in E | x, y \in K\}$, i.e. $E_K = \{(x, y) | x, y \in K\}$.*

**Proof.** Let $x$ and $y$ be two arbitrary nodes from $K$. Since $K$ is a strongly connected component, then there is a directed path from the node $x \in K$ to node $y \in K$ and as $G = G*$ it follows that $(x, y) \in E_K$, whence the theorem.

**Theorem 6.** *Let $G^c$ be the condensed graph of the self-transitive closure graph G = G*. Any directed arc $(K_i, K_j)(i \neq j)$ of the condensed graph must be in the set of arcs $E^{-1}$ of any solution $G^{-1} = (V, E^{-1})$ of the ITC problem, i.e. $\exists x \in K_i$ and $\exists y \in K_j$ so that $(x, y) \in E^{-1}$.*

**Proof.** Let $(G^{-1})^c = \{V', (E^{-1})^c\}$ be the condensed graph of $G^{-1}$. Using the Theorem 4, it is obviously that $((G^{-1})^c)^* = (G^c)^*$ (2). Let $(K_i, K_j)$ be an arbitrary chosen directed arc of the condensed graph $G^c$. Then $(K_i, K_j) \in (E^c)^*$ (3).

Suppose that for any $x \in K_i$ and for any $y \in K_j : (x, y) \notin E^{-1}$. It follows that $(K_i, K_j) \notin (E^{-1})^c$, therefore $(K_i, K_j) \notin ((E^{-1})^c)^*$ (4). From (2), (3) and (4), a contradiction follows.

```
1.    E⁻¹ := Ø;
2.    Find the condensed graph Gᶜ=(V',Eᶜ), where
      V'={Kᵢ|i∈{1,2,...,p}}, Kᵢ={kᵢⱼ∈V|i∈{1,2,...,sᵢ}};
3.    for i:=1 to p do
          if sᵢ>1 then
              for j:=1 to sᵢ do
                  E⁻¹:=E⁻¹ ∪ {(kᵢⱼ, kᵢⱼ₊₁)}; (where k_{i,sᵢ+1} = k_{i1})
              end for;
          end if;
      end for;
4.    E⁻¹:= E⁻¹  ∪ {aᵢ | aᵢ ∈ Eᶜ};
```

*Fig. 5.* **Algorithm 2** (using the condensed graph).

**Theorem 7.** *The algorithm 2 finds an optimal solution $G^{-1} = (V, E^{-1})$ of the inverse transitive closure problem.*

**Proof.** In the step 3 of the algorithm, in the set $E^{-1}$ we introduce directed arcs that form directed elementary cycles of each strongly connected component (which has the minimum number of arcs of all directed cycles). In the step 4, in the set $E^{-1}$ we introduce directed arcs that connects the strongly connected components of the condensed graph. The algorithm finds an optimal solution for the inverse problem of transitive closure ($(G^{-1})^* = G$ and there is no other solution with a less number of arcs) - see Theorems 4, 6.

**Theorem 8.** *The complexity of algorithm 2 is $O(min\{m+n, n{\cdot}p\})$, where m is the number of directed arcs, n is the number of nodes of the self-transitive closure graph G = (V, E), i.e. $m = |E|$, $n = |V|$ and p is the number of strongly connected components of G.*

**Proof.** The complexity of the step 1 of the algorithm is $O(1)$. The complexity of the step 2 is $O(m+n)$ using the depth first search algorithm (DFS). The complexity of the step 3 is $O(n)$, because $s_1+s_2+...+s_p = n$. The complexity of the step 4 is $O(n)$ (there are at most n-1 arcs found in step 2 that connect the strongly connected components, because the condensed graph is a forest of trees). So, the complexity of the whole algorithm is $O(m+n)$. Using the fact that in the self-transitive closure graph G any strongly connected component is complete (Theorem 5), the condensed graph can be also found in step 2 in a complexity of $O(n \cdot p)$ with the algorithm presented in fig. 6.

Being a self-transitive closure graph, the graph G has many arcs (it is "dense"). In most of the cases m is larger than $n \cdot p$. So, it is better to use the algorithm from fig. 6 to find the condensed graph.

## 3. THE NUMBER OF OPTIMAL SOLUTIONS

Here we compute the number of optimal solutions for the ITC problem.

**Theorem 9.** *The number of optimal solutions of the inverse transitive closure problem is $\prod_{i=1}^{p}(s_i-1)!\cdot\prod_{1\le i<j\le p} N_{i,j}$, where $N_{i,j} = s_i\cdot s_j$, if $\exists(K_i, K_j) \in E^c$ or $\exists(K_j, K_i) \in E^c$ and $N_{i,j} = 1$, otherwise.*

**Proof.** All optimal solutions are generated from the condensed graph $G^c = (V, E^c)$. In order to obtain different optimal solutions, the nodes from every strongly connected component must be permuted (noncircular). The noncircular permutations of k order are obtained by holding one position and permuting the other k-1. So, there are (k-1)! noncircular permutations of k order (1).

If the directed arcs which connect the strongly connected components links different nodes from components, different optimal solutions are obtained. Here $N_{i,j}$ is the number of possible different links between the strongly connected components $K_i$ and $K_j$, if there is an arc $(K_i, K_j)$ or $(K_j, K_i)$ that connects the components in the condensed graph (2). So, using (1) and (2), the theorem is proved.

```
r:=0;
p:=1;
select y∈V;
k1,1:={y};
s1:=1;
for x∈V\{y} do
     for i:=1 to p do
          if (x,ki,1)∈E and (ki,1,x)∈E then
                    si:=si+1;
                    ki,si:=x;
          else
               if (x,ki,1)∈E then
                    p:=p+1;
                    sp:=1;
                    kp,1:=x;
                    r:=r+1;
                    ar:=(x,ki,1);
               end if;
               if (ki,1,x)∈E then
                    p:=p+1;
                    sp:=1;
                    kp,1:=x;
                    r:=r+1;
                    ar:=(ki,1,x);
               end if;
          end if;
     end for;
end for;
```

*Fig. 6.* Algorithm for finding the strongly connected components of a self-transitive closure graph.

**Theorem 10.** *If the graph G is (weakly) connected (there is a path between every two nodes of G), then the number of optimal solutions of the inverse problem of transitive closure is* $\prod_{i=1}^{p} s_i! \cdot s_i^{p-2}$

**Proof.** If the graph G is connected then, between every two components $K_i$ and $K_j$ there is an arc $(K_i, K_j)$ or $(K_j, K_i)$ that connects them. This implies that

$$\prod_{1 \le i < j \le p} N_{i,j} = \prod_{1 \le i < j \le p} s_i \cdot s_j = \prod_{i=1}^{p} s_i^{p-1}.$$

It follows (by Theorem 9) that the number of optimal solutions of the ITC problem is

$$\prod_{i=1}^{p} (s_i - 1)! \cdot \prod_{1 \le i < j \le p} N_{i,j} = \prod_{i=1}^{p} (s_i - 1)! \cdot \prod_{i=1}^{p} s_i^{p-1} = \prod_{i=1}^{p} s_i! \cdot s_i^{p-2}.$$