# Swapping Tokens on Graphs


Caio Henrique Segawa Tonetti, Vinicius Fernandes dos Santos & Sebastián Urrutia

Universidade Federal de Minas Gerais, Computer Science Department

caio.tonetti@gmail.com, viniciussantos@dcc.ufmg.br & surrutia@dcc.ufmg.br


## Introduction to the Token Swap (TS) Problem

Let $G = (V, E)$ be a graph with $|V|$ vertices and $|E|$ edges, with distinct tokens placed on it's vertices. The objective is to reconfigure this initial token placement called $f_0 : V \mapsto V$ into the identity token placement $f_i$, that maps every node to itself, through a sequence of pairs of adjacent graph vertices that swap the tokens between these vertices. The aim is to know if it is possible to have a swap sequence $S$ that achieve the objective in $k$ or less swaps, with $k \in \mathbb{N}$.

Applications of the TS problem encompass a wide range of fields. From computing efficient interconnection network structures, [1], computational biology [2, 3], modelling Wireless Sensor Networks (WSS) [4], protection routing [5] to qubit allocation for quantum computers [6, 7].

## Token Swap on Specific Graph Classes

A Conflict Graph $CG_f := (V(G), E_{CG})$ is a digraph that, for a token placement $f$ of a graph $G$, an edge $(u, v) \in E_{CG}$ if and only if $f(u) = v$. Each node has outdegre 1 and the digraph may contain self-loops.

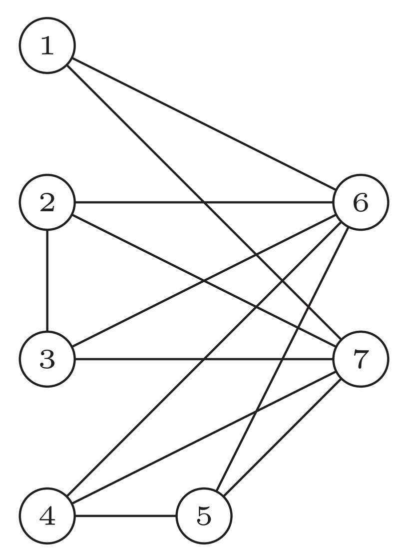


**Figure 1:** Example of a cograph.

A cograph is defined recursively as follows: a graph on a single vertice is a cograph; if $G_1, G_2, \ldots, G_k$ are cographs, then so is their disjoint union; if $G$ is a cograph, then so is its complement $\overline{G}$. A cotree $T(G)$ of a cograph $G = (V, E)$ is a rooted tree representing it's structure. The leaves of $T(G)$ are exactly $V$ and each internal node is either a 0-node and 1-node. The children of an 1-node are 0-nodes or leaves and the children of a 0-node are 1-nodes or leaves. Two vertices are adjacent in a cograph if and only if their lowest common ancestor is an 1-node.

We define $CS(CG_f) = \{C_1, C_2, \ldots, C_k\}$ as the set of permutation cycles of $CG$ for $f$. Let $C^1 \subseteq CS$ be the set of cycles that have a lowest common ancestor of all vertice pairs of $V(C)$ as an 1-node in the cotree or is a cycle of size one and let $C^0 = CS \backslash C^1$. The *Cycle Matching Graph* $H$ of a cograph $G$ has each cycle on $C^0$ as vertice set and two vertices are adjacent if the lowest common ancestor of all vertice pairs in the vertice union in $T(G)$ is an 1-node. Let $\mu(H)$ be the maximum matching in this graph. The following theorem implies the polynomial time solvability of `Token Swap` for cographs.

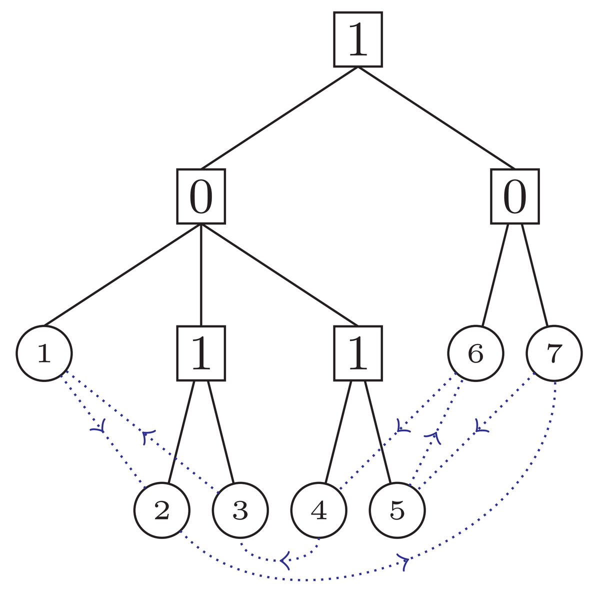


**Figure 2:** Cotree and conflict graph joint representation.

**Theorem.** *Let $G$ be a cograph with an initial token placement $f_0$. The minimum number of required swaps is given by* $|V(G)| + |C^0| - |C^1| - 2 \times |\mu(H)|$.

This result came from two observations: Each independent cycle $C \in CS$ can be solved in $|C| + 1$ or $|C| - 1$ swaps depending on whether this cycle is part of $C^0$ or $C^1$, respectively. Also, it is possible to show that cycle interaction is restricted in the best-case scenario and the best improvement on swaps can be calculated on the value of the maximum matching of the cycle matching graph $H$. This behavior is also being used to find more efficient algorithms in other graph classes like bipartite chain, wheel and gear.

## Token Swap as a Integer Linear Program

The novel TS problem model given by Formulation (1)-(10) tests all possible configurations of the problem with a given upper-bound in the number of swaps $T$, allowing at maximum one swap per step $t \in [T]$. Each step is composed of a set of variables that describe the current configuration, which swap is being selected and Equation 8 checks if a swap sequence solves the current instance. The constant $T$ can be calculated by using any of the best approximation algorithms, or by using the trivial upper-bound $O(n^2)$ on the size of an optimal swap sequence. Binary variables $x_{iut}$ determine if a token $i$ is at node $u$ in step $t$. The binary variables $y_{uvt}$ flags if a swap happened between nodes $u$ and $v$ in step $t$.

$$\min \quad \sum_{\forall uv \in E, u <_M v, \forall t \in [T]} y_{uvt} \tag{1}$$

$$\text{s.t.} \quad \sum_{\forall u \in V} x_{iut} = 1 \qquad \forall_{t\in[T]}, \forall_{i \in V} \tag{2}$$

$$\sum_{\forall i \in V} x_{iut} = 1 \qquad \forall_{t\in[T]}, \forall_{u \in V} \tag{3}$$

$$x_{iut} + x_{ivt+1} \leq y_{uvt} + 1 \qquad \forall_{i\in V}, \forall_{t\in[T-1]}, \forall_{uv\in E}, u <_M v \tag{4}$$

$$x_{ivt} + x_{iut+1} \leq y_{uvt} + 1 \qquad \forall_{i\in V}, \forall_{t\in[T-1]}, \forall_{uv\in E}, u <_M v \tag{5}$$

$$x_{iut} + x_{ivt+1} \leq 1 \qquad \forall_{i\in V}, \forall_{t\in[T-1]}, \forall_{uv\notin E} \tag{6}$$

$$\sum_{\forall u,v \in V, u <_M v} y_{uvt} \leq 1 \qquad \forall_{t\in[T]} \tag{7}$$

$$x_{iiT} = 1 \qquad \forall_{i\in V} \tag{8}$$

$$y_{uvt} \in \{0, 1\} \qquad \forall_{t\in[T]}, \forall_{uv\in E} \tag{9}$$

$$x_{iut} \in \{0, 1\} \qquad \forall_{i\in V}, \forall_{u\in V}, \forall_{t\in[T]} \tag{10}$$

Some techniques are being used in this model to try to achieve a better overall performance, and they will be explained in detail in future papers. The performance measurement, improvements and other models for the problems of `Colored Token Swap` and `Parallel Colored Token Swap` are all planned for future research. Note that the problem of `Parallel Token Swap` currently has a model, but it was omitted here for the sake of conciseness. These models differentiate from the usual TS problem by allowing swaps to be done in parallel or by removing the uniqueness property of a token, assigning a color for a set of tokens instead of a single label.

## References


[1] Fred Annexstein, Marc Baumslag, and Arnold L. Rosenberg. Group action graphs and parallel architectures. *SIAM Journal on Computing*, 19:544–569, 1990.

[2] Vineet Bafna and Pavel A. Pevzner. Sorting by transpositions. *SIAM J. Discret. Math.*, 11:224–240, 1998.

[3] Lenwood Heath and John Vergara. Sorting by short swaps. *Journal of computational biology : a journal of computational molecular cell biology*, 10:775–89, 2003.

[4] Lei Wang and K. W. Tang. The cayley graph implementation in tinyos for dense wireless sensor networks. In *2007 Wireless Telecommunications Symposium*, pages 1–7, Italy, 2007. 2007 Thyrrenian International Workshop on Digital Communication.

[5] Kung-Jui Pai, Ruay-Shiung Chang, and Jou-Ming Chang. Constructing dual-cists of pancake graphs and performance assessment of protection routings on some cayley networks. *The Journal of Supercomputing*, 76:124546, 2020.

[6] Marcos Yukio Siraichi, Vinícius Fernandes dos Santos, Sylvain Collange, and Fernando Magno Quintao Pereira. Qubit allocation. In *Proceedings of the 2018 International Symposium on Code Generation and Optimization*, pages 113–125, New York, NY, USA, 2018. ACM.

[7] Marcos Yukio Siraichi, Vinícius Fernandes dos Santos, Caroline Collange, and Fernando Magno Quintão Pereira. Qubit allocation as a combination of subgraph isomorphism and token swapping. *Proc. ACM Program. Lang.*, 3:1–29, 2019.