Hardness of Subgraph and Supergraph Problems in c-Tournaments

Kanthi K Sarpatwar[1] and N.S. Narayanaswamy[1]

Department of Computer Science and Engineering, IIT madras, Chennai 600036, India
kanthik@gmail.com,swamy@cse.iitm.ac.in

Abstract. Problems like the directed feedback vertex set problem have much better algorithms in tournaments when compared to general graphs. This motivates us to study a natural generalization of tournaments, named c-tournaments, and see if the structural properties of these graphs are helpful in obtaining similar algorithms. c-tournaments are simple digraphs which have directed paths of length at most $c \geq 1$ between all pairs of vertices. We study the complexity of feedback vertex set and r-dominating set in c-tournaments. We also present hardness results on some graph editing problems involving c-tournaments.

Keywords: Parameterized complexity, Feedback vertex set, Dominating set, Graph Modification, Tournaments.

1 Introduction

Literature. In the *directed feedback vertex (arc) set problem*, FVS (resp. FAS) for short, the goal is to delete as few vertices (resp. edges) as possible from a digraph to kill all directed cycles. From the parameterized standpoint (section 2), feedback vertex set can be solved in time $O^*(2^k)$ (In O^*, we ignore the "polynomial" component of the running time) in tournaments (simple, complete digraphs) [1], solution size k being the parameter. On the other hand, the best known result for general feedback vertex set problem is $O^*((1.48k)^k)$ [2]. The question whether there is an algorithm of form $O^*(c^k)$, where c is a constant, for solving feedback vertex set in general digraphs is open.

In the *r-(out)dominating set problem*, we are given a directed graph $G = (V, E)$ and seek a subset of vertices $S \subseteq V$ such that, for every vertex $u \notin S$, there is some vertex $v \in S$ with a directed path of length at most r to u. The 1-dominating set is simply the dominating set problem and while it is known that any tournament has a $\log_2 n$ dominating set, the status of finding a dominating set of size $r \leq \log_2 n$ in tournaments is unknown[1] [3]. Parameterized complexity of this problem is much clearer as it is $W[2]$-complete [4]. A famous theorem by Landau (1955) proves that, in any tournament, the set consisting of a vertex of maximum out degree is a 2-dominating set. Hence, 2-dominating set in tournaments is polynomial time solvable, while it is NP complete(NPC) in general digraphs.

Graph editing problems involve adding or deleting a set of edges or vertices from a graph to make it satisfy certain *property*[2]. Graph editing problems have been extensively studied. For example, editing a graph to make it satisfy a special property called *hereditary* property has been well studied from both classical and parameterized perspective. A property is called hereditary, if for any graph G which satisfies the property, so does every induced subgraph of G. Yannakakis et al. [5] proved that the problem of removing as few edges as possible from a given graph to obtain another graph which satisfies a certain fixed hereditary property is NPC. Raman et al. [6] studied this problem from a parameterized complexity standpoint. Shamir et al. [7] studied the problem of finding a cluster subgraph, which has complete graphs as components. We note that being a cluster graph is hereditary property and so this problem is NPC.

The Idea. We introduce a new class of graphs which bridge the gap between the well structured tournaments and arbitrary digraphs. We call them *c-tournaments*. For a fixed constant c, c-tournaments are digraphs

[1] there is evidence to suggest it is not in **P** and to suggest it is not NP hard

[2] A property is any class of graphs closed under isomorphism.

satisfying the following property, for every pair of vertices u,v, there is a directed path of length at most c connecting them (in at least one direction). Clearly, tournaments are 1-tournaments. In this paper, we study the complexity of the above problems (namely the feedback vertex set, r-dominating set, graph editing problems) and their variants in c-tournaments. The motivation is that the above problems have efficient algorithms in tournaments but not in general graphs and our aim is to understand the complexity of these problems in c-tournaments.

Our Results. In section 2.1, we study the problem of r-dominating set in c-tournaments. We prove that the c-dominating set problem in a c-tournament is $W[2]$-complete. On the other hand, we show that the $2c$-dominating set problem in c-tournaments is polynomial time solvable. In section 3.1, we study the problem of finding a graph obtained by a minimum possible number of edge deletions such that each component is a *2-tournament*. We refer to this as the *2-tournament clustering edge set problem*. We note that being a c-tournament is a *hereditary* property only for $c = 1$. So the NP hardness does not follow. We prove that this problem is NPC, and also show that it is FPT. We then study the problem of 2-tournament completion, that is to add edges to transform a given digraph into a 2-tournament, and prove that this problem is NPC and $W[2]$ complete.

As noted earlier, FVS problem is simpler to solve in tournaments than in general graphs owing to its structure. One natural question that arises is whether the structural properties of c-tournaments would help us solve the FVS (FAS) problem. Our study in section 2.2 answers this question negatively. We lose most of the structural properties helpful to solve FVS problem of tournaments when we reach 3-tournaments. Even 2-tournaments are as hard as general graphs, as we prove that given an $O^*(\alpha^k)$ algorithm, for a constant α, to solve FVS in 2-tournaments, we can solve the problem in general graphs in time $O^*(\alpha^k)$.

2 Complexity Results for c-tournaments

We have used standard graph notation from the book by Bondy and Murty [8] for most part. A vertex's t^{th} order out-degree is the number of vertices which are at an out distance of at most t from it. Similarly the number of vertices from which a vertex can be reached by a directed path of at most t edges is called its t^{th} order in-degree. The sets of those vertices are called the vertex's t^{th} order out-neighborhood and t^{th} order in-neighborhood respectively. We say that a pair of vertices satisfy the property P_c if they are at a directed distance at most c. A digraph in which every vertex pair satisfies P_c is a c-tournament. By a dominating set in a digraph we mean the out-dominating set, unless otherwise specified. We next define the hierarchy of parameterized complexity classes.

Parameterized Complexity: A parameterized problem L is a set of pairs (x, k) such that $x \in \Sigma^*$, where Σ is a finite alphabet and $k \in \mathbb{Z}_+$ is a *parameter*. A parameterized problem L is said to be *fixed parameter tractable* (FPT) if membership in L can be decided in $O(f(k) * n^{O(1)})$, where n is the input size. This class of problems can be thought of as the analogue of P. Note that in this class, for a fixed k, the running time is only polynomially dependent on the input size. When a parameterized problem cannot be shown to be FPT, then an attempt is made to identify its position in a hierarchy of parameterized complexity classes. These classes are analogous to the Π_i hierarchy in the classical complexity, and is referred to as $\mathbf{W}[t]$ hierarchy for $t \in \mathbb{Z}_+$. Completeness for a class in this hierarchy is shown via parameterized reductions. We first define the concept of parameterized reduction or a FPT reduction.

Definition 1. *A parameterized language A FPT-many-one reduces to a parameterized language B if there are a polynomial q, functions $f, g : \mathbb{N} \to \mathbb{N}$ and a Turing machine T such that on any input (x, k), T runs for $f(k)q(|x|)$ steps and outputs $(x', g(k))$ such that $(x, k) \in A \iff (x', g(k)) \in B$.*

In the natural hierarchy of classes of parameterized languages it is known that $FPT \subseteq W[1] \subseteq W[2] \cdots \subseteq W[poly]$. It has been shown, see [9, 10], that dominating set is complete for the class $W[2]$, and clique is complete for the class $W[1]$ under FPT reductions. For further details and the definition of $W[t]$ we refer the reader to the texts on parameterized complexity by Downey and Fellows [9] and Niedermeier [10].

2.1 r-dominating set in c-tournaments

Following is the formal definition of the problem *r-dominating set*, for every fixed $r \in N$.
Instance A directed graph $G = (V, E)$, $k \in N$
Solution $V' \subseteq V$ such that $|V'| \leq k$ and $\forall u \in V\ \exists v \in V'$, such that, there is a directed path of length at most r from v to u.
As noted the problem of 1-dominating set is $W[2]$ hard in tournaments but the problem of r-dominating set is in **P**, for $r \geq 2$. In this section we observe that c-tournaments carry some structure of tournaments which allows us to solve r-dominating set for $r \geq 2c$. We prove that the problem of c-dominating set is $W[2]$ complete in c-tournaments. We leave open the status of $(c + i)$-dominating set in c-tournaments, for $1 \leq i \leq c - 1$. First we extend Landau's theorem in tournaments to c-tournaments.

Theorem 1. *Let T be a c-tournament and v a vertex with a maximum number of vertices at a distance of at most c. Then $\{v\}$ is a $(2c)$-dominating set of T.*

Proof. Let $N_c(v)$ denote the c^{th} order out neighborhood of v. Let u be a vertex not in $N_{2c}(v)$. Since T is a c-tournament, it follows that every vertex in $N_c(v) \cup \{v\}$ is in $N_c(u)$. This is a contradiction to the fact that v is a vertex with the largest c^{th} order neighborhood. Hence, the lemma. □

Thus the problem of r-dominating set is polynomial time solvable for $r \geq 2c$. We next show that c-dominating set in c-tournament is $W[2]$ complete. The reduction is from dominating set in tournaments [4].

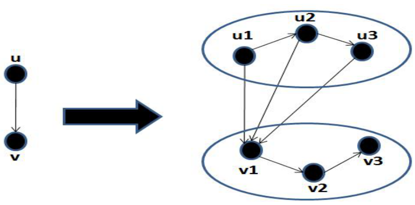

Fig. 1. Reduction of dominating set in tournaments to 3-dominating set in 3-tournaments

Theorem 2. *The c-dominating set problem restricted to c-tournaments is W[2] complete, where the parameter of interest is the size of the desired c-dominating set.*

Proof. We reduce the dominating set problem in tournaments to the c-dominating set problem in c-tournaments (refer to figure 1). Given a tournament $T = (V, E)$, we construct a c-tournament $T_c = (V_c, E_c)$ as follows: each vertex $v \in V$ is split into c copies $v_1, v_2, \dots v_c$ which are all connected by a $c-1$ path directed by adding edges (v_i, v_{i+1}), for all $1 \leq i \leq c - 1$. If $(u, v) \in E$, we add edges $\{(u_i, v_1) | 1 \leq i \leq c\}$. We now show that T_c is a c-tournament with a c-dominating set of size at most k if and only if T has a dominating set of size at most k. This proves the $W[2]$-hardness.
Firstly, we show that T_c is a c-tournament by doing case analysis and exhibiting paths of length at most c in each case:

- For any pair of vertices of the form $v_i, v_j \in V_c \ni i < j : v_i v_{i+1} \dots v_j$ is a connecting path of length at most c.
- For any pair of vertices of the form $u_i, v_j \in V_c \ni (u, v) \in E$: $u_i v_1 v_2 \dots v_j$ is the required path of length at most c.

Secondly, if D is a dominating set of V with size k, then $D_c = \{u_1 : u \in D\}$ is clearly a c-dominating set of T_c. Indeed, every vertex $u_j \in V_c, j \in [1, c-1]$, is c-dominated by the corresponding vertex u_1 due to the path $u_1 u_2 \dots u_j$. This shows that if T has a dominating set of size k, then T_c has a c-dominating set of size k.
Next, for any c-dominating set D_c of T_c, define $D = \{v | v_i \in D_c, \text{ for some } i, 1 \le i \le c\}$. By the definition of D, for every $v \in D$ there is at least one $v_i \in D_c$. Hence, the cardinality of D is at most the cardinality of D_c. Finally we observe that D is a dominating set of T: Indeed, every vertex u is dominated by the vertex v in T where u_c is dominated by v_c in T_c. Thus, if there is a c-dominating set of T_c with size at most k, then there is a dominating set of T with size at most k. Hence, the theorem follows. □

We conclude this sub-section by mentioning that c-dominating set problem in c-tournaments is closely related to dominating set problem in tournaments. Given a c-tournament T_c, we construct the following tournament T: $V(T) = V(T_c)$, and for each pair $\{u, v\}$, we add an edge from (u, v) in $T \iff$ there is a path of length at most c between u and v in T_c. If there is such a path in both directions, we arbitrarily pick one edge. Clearly, T is a tournament and a dominating set in T is a c-dominating set in T_c. As there is a dominating set of size $\log_2 n$ in a tournament, there is a c-dominating set of size $\log_2 n$ in a c-tournament. This gives us an algorithm to solve the minimum c-dominating set problem in c-tournaments in $O(n^{\log_2 n})$ time.

2.2 Complexity of FVS and FAS in c-tournaments

First we prove that the FVS and FAS problems in 3-tournaments are equivalent to the FVS and FAS problems in general directed graph. Given an instance G of FVS (FAS), add two new vertices, an *in-vertex* t and an *out-vertex* s. The vertex s has edges directed out of it to each vertex of G. Similarly t has edges into it from each vertex of G. Finally we add an edge from t to s. Let the resulting graph be G'. G' is a 3-tournament on $|V(G)| + 2$ vertices, and can be constructed in polynomial time. Indeed, consider a pair of vertices u, v in $V(G')$. If either of the vertices is s or t, then the pair of vertices is an element of $E(G')$. If neither is s or t, then there is a path from u to v formed by the three edges $(u, t), (t, s), (s, v)$. Therefore, G' is a 3-tournament. The following theorem connects the FVS (FAS) of G and G'.

Theorem 3. *G has a FVS (FAS) set of size $k \iff$ the 3-tournament G' has a FVS (FAS) set of size $k + 1$.*

Proof. We first prove the claim for the FVS case. In the case when G has a FVS of size k, say S, define $S' = S \cup \{s\}$. We prove, by contradiction, that $G' \setminus S'$ is acyclic. A cycle in $G' \setminus S'$ must contain at least one of s or t, since $G \setminus S$ is acyclic. Since $s \in S'$, such a cycle must contain t. This is impossible since t has no out-edge in $G' \setminus S'$. Therefore, S' is FVS of G' and $|S'| = k + 1$. In the other direction we consider the case when G' has a FVS of size $k + 1$, say S'. By the construction of G', $S' \cap \{s, t\} \neq \phi$, since every vertex $v \in V(G)$ is in a cycle formed by the edges $(v, t), (t, s), (s, v)$. Therefore, the set $S' \cap V(G)$ is a FVS of G, and is clearly of cardinality at most k. The proof for the feedback arc set is essentially the same argument based on the observation that any feedback arc set of G' must contain the arc (t, s) and it is sufficient to add (t, s) to any feedback arc set of G to get a feedback arc set of G'. □

It might be misconstrued that since a 3-tournament can be obtained by adding just 2 vertices to any graph, they are not very restrictive. The 3-tournaments have enough structure to solve the 6-dominating set problem easily, which is $W[1]$ complete in general graphs.

We now prove that solving FVS problem in 2-tournaments is as hard as solving them in general simple directed graphs using a more involved construction.
The Reduction: Given a simple directed graph on n vertices G, we construct a 2-tournament T as following (refer figure 2): T comprises of 3 graphs G, G', and G''. We assign distinct labels to the n vertices of G using $\lceil \log_2 n \rceil$ bits. G' is a graph on $\lceil \log_2 n \rceil$ new vertices $u_1, \dots, u_{\lceil \log_2 n \rceil}$, and $(u_i, u_j) \in E(G)$ for each pair $1 \le i < j \le \lceil \log_2 n \rceil$. For each $1 \le i \le \lceil \log_2 n \rceil$, and $v \in V(G)$ the edge (u_i, v) is present if the i-th bit in the label of v is 0. Otherwise the edge (v, u_i) is added. G'' is a graph that has two copies, T^+ and T^-, of a

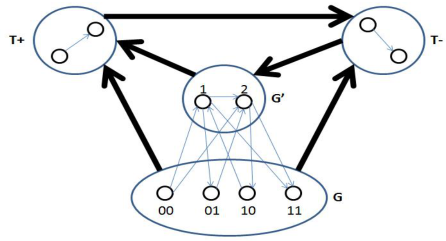

Fig. 2. Converting a graph on 4 vertices to a 2-tournament. The thick lines between any two groups of vertices represent edges between every pair of vertices, one from each group.

transitive tournament on $\lceil \log_2 n \rceil$ vertices. To obtain T, we add edges from all vertices of G into all vertices of G'', from all vertices of G' into all vertices of T^+, from all vertices of T^+ into all vertices of T^-, and from all vertices of T^- into all vertices of G'. It is easy to observe that T is a 2-tournament.

Lemma 1. *In the above construction, G has a FVS of size k if and only if T has a FVS of size $k + \lceil \log_2 n \rceil$*

Proof. If G has a FVS of size k, say S, then $S \cup G'$ is clearly a FVS of size $k + \lceil \log_2 n \rceil$ of T. Now if S' is a FVS of T, then we claim that S' contains atleast $\lceil \log_2 n \rceil$ vertices of $G'' \cup G'$. Indeed, there are at least $\lceil \log_2 n \rceil$ disjoint triangles in the subgraph of T induced by vertices of $G'' \cup G'$ (simply split $G'' \cup G'$ into triplets of vertices, each triplet consisting of exactly one vertex from each of G', T^+ and T^-). At least one vertex from each of these triangles must be present in every FVS of T. Hence, the claim. Thus, if T has a FVS of size $k + \lceil \log_2 n \rceil$, it must have a FVS of size at most k in G. Hence, the lemma. □

The following theorem is an easy consequence of the above lemma.

Theorem 4. *An algorithm to test if a 2-tournament has a FVS of size at most k in $O^*(2^{O(k)})$ time can be used to test if a directed graph has a FVS of size at most k in time $O^*(2^{O(k)})$.*

3 2-tournament Subgraph and Supergraph Problems

This section is devoted to the study of some variations of graph editing problems involving 2-tournaments. As mentioned earlier, the clique clustering problem is well studied and is known to be NPC and FPT. We now formally define the clique clustering problem and its variation, studied in the next subsection. In the following, we say that a subgraph of a directed graph is a component if and only if the underlying undirected subgraph is a connected component of the underlying undirected graph.

Clique clustering.
Instance A simple undirected graph and $k \in \mathbf{N}$.
Solution Remove at most k edges such that each component of the resulting graph is a clique.

2-tournament clustering.
Instance A simple directed graph and $k \in \mathbf{N}$.
Solution Remove at most k edges such that each component of the resulting graph is a 2-tournament.

3.1 2-tournament clustering edge set is NPC and FPT

In the following sections, we say C is a component of digraph G, if C is a component in the underlying undirected graph of G. We show that 2-tournament clustering edge set is NPC by a reduction from the clique clustering problem [7] on undirected graphs. We subsequently show that 2-tournament clustering edge set is FPT.

Reduction Given an undirected graph $G(V, E)$, construct the following graph $G'(V', E')$ where $V' = \{v^+, v^- : v \in V\}$ and $E' = \{(v^+, u^-), (u^+, v^-), (v^-, v^+), (u^-, u^+) : \{v, u\} \in E\}$. Throughout this section, the graph G' is used to denote the graph obtained from this reduction. Also, we refer to v^+ and v^- as complements of each other.

Lemma 2. *G is a clique $\iff$ G' is a 2-tournament.*

Proof. $\Rightarrow$ G is a clique. Since G is a clique, any two vertices of opposite sign in G' are adjacent. Any two vertices of the same sign, say u^+ and v^+ are connected by a path of length 2 involving the edges $\{u^+, v^-\}, \{v^-, v^+\}$. Therefore, G' is a 2-tournament.
$\Leftarrow$ G' is a 2-tournament. Let $\{u, v\} \in G$ be two non-adjacent vertices. In G', we know that there is a directed path of length at most 2 connecting u^+ and v^+. WLOG[3] let the path be from u^+ to v^+. We observe that since there is no edge in G' between two vertices of same sign, the path must be of length exactly 2. Also since the only edge into v^+ is v^-v^+, the 2-path must be $\{u^+, v^-\}, \{v^-, v^+\}$. Hence $u^+v^- \in E(G')$ which in turn implies $\{u, v\} \in E(G)$ - a contradiction to the assumption that u and v are non-adjacent. Therefore, if G' is a 2-tournament, it follows that G is clique. □

In a directed graph G, let C denote a 2-tournament cluster, that is a subgraph in which each component (of the underlying undirected graph) is a 2-tournament. The edge set $E(G) \setminus E(C)$ is called a *clustering edge set*, and the set $V(G) \setminus V(C)$ is called a *clustering vertex set*.

Lemma 3. *Let C be a 2-tournament that is a subgraph of G'. There is at most one vertex $v^+ \in V(C)$ such that $v^- \notin V(C)$. Similarly, there is at most one $u^- \in V(C)$ such that $u^+ \notin V(C)$.*

Proof. Proof is by contradiction. Let $w^+, v^+ \in V(C)$ such that both $w^-, v^- \notin V(C)$. Since C is a 2-tournament, there is a path of length at most 2 connecting the vertices $\{v^+, w^+\}$. Since there is no edge between $\{v^+, w^+\}$, the path must have length exactly 2. WLOG, let the path be from v^+ to w^+. Since the only incoming edge into w^+ is from w^- we must have w^- in the path from v^+ to w^+. This implies $w^- \in V(C)$ which is a contradiction to our assumption. Hence, the proof. □

Lemma 4. *Let M be a 2-tournament clustering edge set for G'. There is a 2-tournament clustering edge set M' such that $|M'| \leq |M|$, and each pair $\{v^+, v^-\}$ occurs in one component in $G' \setminus M'$.*

Proof. If each pair $\{v^+, v^-\}$ occurs in a component in $G' \setminus M$, then M is the desired M' and the statement of the lemma is true. Let us assume that this is not the case. Let v^+ and v^- be in different components C_1 and C_2 of $C = G' \setminus M$, respectively. We first show that either moving v^+ to C_2 or moving v^- to C_1 yields a 2-tournament clustering edge set M'' such that $|M''| \leq |M|$.
First, we prove that placing v^+, v^- in the same component (adding back and deleting edges appropriately) will give another 2-tournament cluster. Without loss of generality, let us place these in C_1. Let C_2 cease to be a 2-tournament. Let u and w be two (signed) vertices whose property P_2 is destroyed. Clearly u and w must form a 2-path with v^-, as the latter's removal caused the destruction of P_2 for these vertices. So both, u and w must have $'+'$ sign and one of them must be a complement of v^- i.e., one of them must be v^+, which is impossible. A similar argument shows C_1 remains a 2-tournament on addition of v^-. Hence, the new graph is a 2-tournament cluster. Let c_1 be the number of vertices in $C_1 \setminus \{v^+\}$ which form an edge with v^- in G' and let c_2 be the number of vertices in $C_2 \setminus \{v^-\}$ that form an edge with v^+ in G'. In C_1 the in-degree of v^+ is zero, and in C_2 the out-degree of v^- is zero. Let d_1 and d_2 denote out-degree of v^+ in C_1 and in-degree of v^- in C_2, respectively. By applying lemma 3, it follows that there can be at most one other vertex in C_1 whose complement is not in C_1. Therefore, it follows that $c_1 \geq d_1 - 1$. By a symmetric argument, it follows that $c_2 \geq d_2 - 1$. The clustering edge set corresponding to the transfer of v^+ to C_2 has $|M| - c_2 - 1 + d_1$ edges and this is at most $|M| + d_1 - d_2$. Similarly, moving v^- from C_2 to C_1 results in

[3] Without loss of generality

a clustering edge set of cardinality $|M| + d_2 - c_1 - 1$ which is at most $|M| + d_2 - d_1$. Therefore, it follows that one of the two moves yields a edge clustering set M'' whose cardinality is at most $|M|$, and in $G \setminus M''$ v^+ and v^- are in the same component. Performing this move for all pairs u^+ and u^- which are in different components in C, results in the desired M', and proves the lemma.

□

Using lemma 4, we now complete a reduction of finding a clique clustering edge set to finding a 2-tournament clustering edge set.

Theorem 5. *For each $k \geq 0$, an undirected graph G has a clique clustering edge set of size at most k if and only if G' has a 2-tournament clustering edge set of size at most $2k$.*

Proof. $\Rightarrow$: Let M be a clique clustering edge set of G such that $|M| = k$. Let $M' = \{(u^+, v^-), (v^+, u^-) | \{u, v\} \in M\}$. Clearly $|M'| = 2k$ and $|M'| \subseteq E(G')$, and by lemma 2 each component of $G' \setminus M'$ is 2-tournament. Therefore, M' is a 2-tournament edge clustering set.
$\Leftarrow$ By lemma 4, let M' be an inclusion-minimal 2-tournament clustering edge set such that $|M'| \leq 2k$ and each pair $\{v^+, v^-\}$ is in a single component of $G' \setminus M'$. We claim that if $(x^+, y^-) \in M'$, then $(y^+, x^-) \in M'$. Let $(x^+, y^-) \in M'$ and $(x^-, y^+) \notin M'$. There is a component $C \in G' \setminus M'$ such that C is a 2-tournament and $x^-, y^+ \in C$. This implies, by the choice of M', vertices $x^+, y^- \in C$. We observe that adding the edge (y^+, x^-) to $G' \setminus M'$ will result in addition of the edge to the component C, leaving every other component untouched. It is evident that if C is a 2-tournament, then so is $C \cup (x^+, y^-)$ (adding a new edge will not destroy any 2-path). Hence, $G' \setminus M''$ is a cluster of 2-tournaments, where $M'' = M' \setminus (y^+, x^-)$. Hence M'' is a 2-tournament edge clustering set with $M'' \subset M'$, contradicting the minimality of M'. Now, we define $M \subseteq E$ as the set $M = \{\{u, v\} | (u^+, v^-), (v^+, u^-) \in M'\}$. Since $|M'| \leq 2k$, it follows that $|M| \leq k$ and by lemma 2 each component of $G \setminus M$ is a clique. Therefore, M is a clique clustering edge set of size at most k. Hence, the theorem. □

We have reduced the clique clustering edge set problem to *2-tournament* clustering edge set problem. Since the former is NP hard (NPH), so is the latter, and its membership in NP is also straightforward. Therefore, the 2-tournament clustering edge set problem is NPC. Using the same reduction it follows that deciding if a directed graph has a 2-tournament of size at least k is both NPC and W[1] hard. The reduction is from the clique problem, which is NPC and W[1] hard, and the transformation is exactly the same as one outlined above. From lemma 2 we have that there is a clique of size k in G if and only if G' has a 2-tournament of size $2k$.
2-tournament clustering edge set is FPT: We present an FPT algorithm to find out if a given directed graph has a 2-tournament clustering edge set of size at most k. The algorithm is based on lemma 5 which is used to show that if a directed graph is not a cluster of 2-tournaments, then there exist two vertices, which do not have a directed 2-path between them but are at a distance of at most 3 in the underlying undirected graph.

Lemma 5. *Let G be a directed graph which is not a 2-tournament such that the underlying directed graph is connected. There exist two vertices for which the distance in the undirected graph is at most 3 but are not at directed distance 2 in G.* □

Proof. Let us assume the contrary. Let S denote the set of vertex pairs in G whose minimum distance in G, in either direction, is more than 2. Let Let $\{u_{min}, v_{min}\}$ be a pair in S such that the distance between u_{min} and v_{min} in the underlying undirected graph is the least over all pairs in S. Let this distance be d. If we assume that the statement of the lemma is wrong, then $d \geq 4$. Let the shortest undirected path be $P_{min} = \{u_{min}, v_1, v_2, v_3, ..., v_{min}\}$. By our assumption($d \geq 4$), the pair $\{u_{min}, v_3\}$ is not an element of S. Therefore, there is a path in the directed graph between u_{min} and v_3 of length at most 2(let w be the connecting vertex). This now gives us a path shorter than P_{min} between u_{min} and v_{min} in the underlying undirected graph(in both the cases, when the vertex w is in P_{min} or not), a contradiction to the fact that P_{min} is a path of shortest length. Therefore, our assumption is wrong. Hence, the lemma. □

Algorithm: Lemma 5 is converted into an algorithm based on the following observation: if each component is a 2-tournament, we are done. If not, consider a vertex pair (u, v) in a component at a distance more than 2 in the directed graph, and u and v are a distance at most 3 in the underlying undirected graph. Existence of such a pair is guaranteed by Lemma 5 and can be found in polynomial time. Let P_{uv} be any such path in the underlying undirected graph. By definition, a 2-tournament clustering edge set M ensures that u and v are in different components in $G \setminus M$. Therefore, M must contain at least one directed edge corresponding to the edges in P_{uv}. This gives us a simple "bounded search tree" - we branch on the edges of P_{uv}, i.e., include them into the potential solution one at a time. Deleting the selected edge from the graph in each case, we look for a $k-1$ solution in the resultant graph. Clearly, the running time of the algorithm is $O^*(3^k)$(The running time satisfies the recurrence $T(k) \leq T(k-1) * 3 + p(n)$).

3.2 Digraph completion problems related to 2-tournaments

A natural question is to study the complexity of finding a set of at most k edges to be added to a digraph so that the resulting digraph becomes a 2-tournament. This is the 2-tournament completion problem, and we prove hardness results for this problem using the following problem:

Problem 1. Single vertex satisfaction(SVS)
Given: A simple directed graph $G = (V, E)$,a vertex $v \in V$, integer $k \geq 0$.
Question: Add at most k edges so that for all $u \in V(G)$, there exists a path of length at most 2 connecting u and v.

To prove hardness of this problem, we transform an instance of the dominating set in a bipartite graph to an instance of SVS. Dominating Set in bipartite graphs is NPC and, in the parameterized context, $W[2]$ hard.

Theorem 6. *SVS is NPC and $W[2]$ hard.*

Proof. **Reduction:** Given an instance $G = (A, B)$, an integer $k \geq 0$, of dominating set in bipartite graphs, we construct a digraph G' by adding a new vertex v to G, and then by orienting all edges from the set A to B. We now show that G has a dominating set of size at most k if and only if there is a set of at most k edges that can be added to G' so that for all $u \in V(G') \setminus \{v\}$, there exists a directed path of length at most 2 connecting u and v. If D is a dominating set of G, then $M = \{(v, u) : u \in D \cap A\} \cup \{(u, v) : u \in D \cap B\}$ is clearly a solution of size k for the SVS problem instance (G', v). In the other direction, if M is a solution of size k for the SVS problem instance (G', v), we prove that there is a dominating set of size at most k of G. First we note a simple property of dominating sets in a graph: adding m edges to a graph decreases its minimum dominating set size by at most m. Let $M' \subseteq M$ be the set of edges with both ends in G (note that G does not include v). From the above property, if d' is the minimum dominating set size of $G \cup M'$ then $d' \geq d - |M'|$, where d is the minimum dominating set size of G. Now, the edges of $M \setminus M'$ must be incident on v. We claim that $|M| - |M'| \geq d'$. Let S be the set of vertices adjacent to v. It is clear that S must be a dominating set of $G \cup M'$, since there is a directed path of length at most 2 between v and every vertex in G. Thus $d' \leq |S| = |M| - |M'|$. Thus we have $k = |M| \geq d' + |M'| \geq d$. This shows that the size of the minimum dominating set is of size at most k. This completes the reduction of the dominating set in bipartite graphs to the SVS problem. □

2-tournament completion: We now show that the problem of adding as few edges as possible to a general directed graph to make it a *2-tournament* is NPC and W[2] hard. We reduce the problem of SVS to this problem. Given a general directed graph $G = (V, E)$ and a vertex $v \in V$, we construct the following graph $G' = (V', E')$.

Construction:

$$V' = V \cup V_1 \cup \{v_{ex}\}$$
$$V_1 = \{v_{u,w} : \forall \{u, w\} \in V \setminus \{v\}\} \quad (1)$$

$$
\begin{aligned}
E' = E &\cup \{(u, v_{u,w}), (v_{u,w}, w), \forall v_{u,w} \in V_1\} \\
&\cup \{(u, v_{ex}), \forall u \in V\} \cup \{(v_{ex}, v_{u,w}), \forall v_{u,w} \in V_1\} \\
&\cup \{(u, v), \forall \{u, v\} \in V_1\}
\end{aligned} \tag{2}
$$

Theorem 7. *2-tournament completion is NPC and W[2] hard.*

Proof. We present a polynomial time transformation of an instance of SVS to an instance of the 2-tournament completion. Let $(G, v, k \geq 0)$ be an instance of SVS. Let G' be the graph constructed from the above instance $(G, v, k \geq 0)$. We show that there is a solution of size at most k to the SVS instance if and only if there is a solution of size at most k for the 2-tournament completion on the graph G'.
Let M, a set of edges such that $|M| \leq k$, be a solution of SVS problem for the instance $I = (G, v, k)$. We prove that M is also a solution of the problem 2-tournament completion for the instance G', that is, we prove that $G' \cup M$ is a 2-tournament. We prove this by showing a path of length at most 2 between every pair of vertices, $\{u, w\} \subseteq V'$. By construction, in G' there is a path of length at most 2 between all pairs except when $u \in V$ and $w = v$, and for this case since M is a solution of SVS problem of (G, v, k), there is path of length 2 connecting u and w. Therefore, it follows that $G' \cup M$ is a 2-tournament.
We now prove that a set of at most k edges, M, added to G' to make it a 2-tournament, immediately yields a solution to the SVS problem on the instance (G, v, k). We note that there are four possibilities for any edge of M,

1. It is directed from v to a vertex $v_{u,w} \in V_1$
2. It is directed from $v_{u,w} \in V_1$ to v
3. It is directed from $u \in V \setminus \{v\}$ to $v' \in V_1$
4. It is directed from $v' \in V_1$ to $u \in V \setminus \{v\}$
5. Both ends of the edge are in V

Given M, we construct M' in the following way. For each edge in M, we do the following, according to the above cases:

1. In this case, add an edge (v, w) to M'
2. Add edge (u, v) to M'
3. Add edge (u, v) to M', if not added already.
4. Add edge (v, u) to M', if not added already.
5. Add the edge to M'

We claim that $G' \cup M'$ is a 2-tournament and M' is also a solution of SVS instance (G, v, k). Since it is clear that $M' \leq M$, we will have our theorem. Let the former claim be false. The only way this can happen is when there exist $x = v, y \in V$ such that there is no 2-path or direct edge between them in $G' \cup M'$. They must have a 2-path in $G' \cup M$, by the definition of M. We prove that this is impossible by enumerating all cases and by showing contradictions:

1. The 2-path (in $G' \cup M$) is of the form $(v, v_{u,w}, y)$ or $(y, v_{u,w}, v)$, where $v_{u,w} \in V_1$. At least one of the edges of the 2-path must be in M, otherwise there will be a 2-path in G' and hence in $G' \cup M'$. All possibilities in this case will result a direct edge between v and y in M'. This is a contradiction to the fact that there is no direct edge between v and y in $G' \cup M'$.
2. The 2-path is of the form (v, z, y) or (y, z, y) where $z \in V$ or there is a direct edge between v and y in $G' \cup M$. We note that all these edges are, by construction, present in M', which is again a contradiction.

Hence, $G' \cup M'$ is a 2-tournament. We also note that all paths in $G' \cup M'$ required to satisfy the property P_2 of v, y are contained in V and hence M' is a solution of the SVS instance (G, v, k), as $|M'| \leq |M| \leq k$. Hence, the claim and the theorem. □

4 Conclusion

A natural generalization of tournaments, r-tournaments, is the focus of current article. Hardness of some problems which are easy to solve in tournaments but "hard" to solve in general digraphs is studied in r-tournaments. These graphs seem to carry on some structure of tournaments, when we look for a r-dominating set. The structure, however, does not help us solve the FVS problem: given efficient algorithm to solve 2-tournament FVS, we can obtain efficient algorithms for general digraph FVS. We also studied a generalization of cluster graph editing problem and obtained efficient FPT algorithms. Finally we consider an interesting graph augmentation problem, 2-tournament completion and prove that this problem is NP hard and $W[2]$ hard. As an intermediate step to proving the NP hardness of this problem, we prove the hardness of another graph augmenting problem named Single Vertex Satisfaction . It would be very interesting to see if other NPC problems, like the longest path problem, have efficient algorithms in r-tournaments. While it is known that tournaments always have a hamiltonian path, 2-tournaments need not have one (it is easy to construct a counter example). An interesting question would be finding out if the longest path problem is polynomial time solvable or proving that it is NPC. Other interesting questions would be finding out the status of r-dominating set in c-tournaments, where $r \in [c+1, 2c-1]$. In conclusion, r-tournaments seem to be an interesting class of graphs. The study of hard problems in r-tournaments could help in obtaining better algorithms in general graphs, as these graphs form a natural bridge between two extremes: tournaments and general connected digraphs.

References

1. M. Dom, J. Guo, F. Hüffner, R. Niedermeier, and A. Truss, "Fixed-parameter tractability results for feedback set problems in tournaments," *J. of Discrete Algorithms*, vol. 8, pp. 76–86, March 2010.
2. J. Chen, Y. Liu, S. Lu, B. O'Sullivan, and I. Razgon, "A fixed-parameter algorithm for the directed feedback vertex set problem," in *STOC '08: Proceedings of the 40th annual ACM symposium on Theory of computing*, (New York, NY, USA), pp. 177–186, ACM, 2008.
3. N. Megiddo and U. Vishkin, "On finding a minimum dominating set in a tournament (note)," *Theor. Comput. Sci.*, vol. 61, no. 2-3, pp. 307–316, 1988.
4. R. G. Downey and M. R. Fellows, "Parameterized computational feasibility," in *Feasible Mathematics II*, pp. 219–244, Birkhauser, 1994.
5. M. Yannakakis, "Node-and edge-deletion np-complete problems," in *STOC '78: Proceedings of the tenth annual ACM symposium on Theory of computing*, (New York, NY, USA), pp. 253–264, ACM, 1978.
6. S. Khot and V. Raman, "Parameterized complexity of finding subgraphs with hereditary properties," 2002.
7. R. Shamir, R. Sharan, and D. Tsur, "Cluster graph modification problems," *Discrete Appl. Math.*, vol. 144, no. 1-2, pp. 173–182, 2004.
8. J. Bondy and U.S.R.Murty, "Graph theory, graduate texts in mathematics," *Macmillan*, 1976.
9. R. G. Downey and M. R. Fellows, "Parameterized complexity," *Springer-Verlag*, 1999.
10. R.Niedermeier, "An invitation to fixed parameter algorithms," *Oxford University Press*, 2006.