Iranian Journal of Mathematical Sciences and Informatics
Vol. 10, No. 2 (2015), pp 115-122

On the Computational Complexity of the Domination Game

Sandi Klavžar [a,b,c*], Gašper Košmrlj [d], Simon Schmidt [e]

[a] Faculty of Mathematics and Physics, University of Ljubljana, Slovenia.
[b] Faculty of Natural Sciences and Mathematics, University of Maribor, Slovenia.
[c] Institute of Mathematics, Physics and Mechanics, Ljubljana, Slovenia.
[d] Abelium, R&D d.o.o., Ljubljana, Slovenia.
[e] Institut Fourier, SFR Maths à Modeler, Joseph Fourier's University, Grenoble, France.

E-mail: sandi.klavzar@fmf.uni-lj.si
E-mail: gasper.kosmrlj@student.fmf.uni-lj.si
E-mail: simon.schmidt@ujf-grenoble.fr

Abstract. The domination game is played on an arbitrary graph G by two players, Dominator and Staller. It is known that verifying whether the game domination number of a graph is bounded by a given integer k is PSPACE-complete. On the other hand, it is showed in this paper that the problem can be solved for a graph G in $\mathcal{O}(\Delta(G) \cdot |V(G)|^k)$ time. In the special case when $k = 3$ and the graph G considered has maximum diameter, the complexity is improved to $\mathcal{O}(|V(G)| \cdot |E(G)| + \Delta(G)^3)$.

Keywords: Domination game, Game domination number, Complexity of algorithms.

2000 Mathematics subject classification: 05C57, 91A43, 05C69.

1. Introduction

The domination game is played on an arbitrary graph G by *Dominator* and *Staller*. They are taking turns choosing a vertex from G such that at

*Corresponding Author

Received 01 July 2015; Accepted 07 July 2015

least one previously undominated vertex becomes dominated. The game ends when no move is possible, the score of the game being the total number of vertices played. Dominator wants to minimize the score, while Staller wants to maximize it. The game is called *D-game* when Dominator starts it, and *S-game* if Staller has the first move. Assuming that both players play optimally, the *game domination number* $\gamma_g(G)$ (the *Staller-start game domination number* $\gamma_g'(G)$) of a graph G, denotes the score of D-game (S-game, resp.).

The game was introduced in 2010 in [4] and received a considerable attention afterwards. A strong motivation factor for the game is the 3/5-conjecture posed and studied in [12], and further investigated in depth in [2, 5, 6, 9]. Additional results and aspects of the domination game were also investigated. For instance, guarded subgraphs and their role in the game was studied in [3], the behaviour of the game on the disjoint union in [8], realizability of game domination numbers in [14], and extremal trees with respect to the game in [15]. We also mention that two closely related games were introduced very recently: the total domination game [10, 11] and the disjoint domination game [7].

In this paper we are interested in the complexity point of view of the game, especially motivated by the result from [1] asserting that verifying whether the game domination number of a graph is bounded by a given integer k is PSPACE-complete. To put the game into another perspective we observe in the next section that the problem can be solved for a graph G in $\mathcal{O}(\Delta(G)\cdot|V(G)|^k)$ time. This means that if k is *not* part of the input, the problem becomes polynomial. Then, in Section 3, using a characterization from [13], we show that the general complexity $\mathcal{O}(\Delta(G)\cdot|V(G)|^3)$ can be improved to the time $\mathcal{O}(|V(G)|\cdot|E(G)|+\Delta(G)^3)$ within the class of graphs of diameter 6.

In the rest of the section we introduce some additional concepts and notation needed. A *partially-dominated graph* is a graph together with a declaration that some vertices are already dominated. Such vertices thus need not be dominated in the rest of the game. If $S\subseteq V(G)$, then the partially dominated graph in which vertices from S are already dominated will be denoted by $G|S$. As usual, if x is a vertex of G, then its open and closed neighborhood will be denoted by $N(x)$ and $N[x]$, respectively. If G is a graph, then $S_r(x)=\{y\in V(G):\ d_G(x,y)=r\}$ is the sphere with center x and radius r, and $B_r(x)=\{y\in V(G):\ d_G(x,y)\le r\}$ is the ball with center x and radius r. Finally, the maximum degree of G is denoted by $\Delta(G)$; we will simply write Δ when G will be clear from the context.

2. On the Complexity of the Game

The D-game domination problem is the following.

D-Game Domination Problem	
Input:	A graph G, and an integer k.
Question:	Is $\gamma_g(G) \leq k$?

Similarly, The S-game domination problem is:

S-Game Domination Problem	
Input:	A graph G, and an integer k.
Question:	Is $\gamma_g'(G) \leq k$?

As already mentioned, it was proved in [1] that these problems are log-complete in PSPACE. On the other hand, we show now that the following holds.

Theorem 2.1. *If G is a graph of order n and k is a fixed integer, then the D-game domination problem and the S-game domination problem can be solved in $\mathcal{O}(\Delta(G)n^k)$ time.*

Proof. We jointly define two recursive algorithms, A and A′, for D-game and for S-game, as follows.

```
Algorithm  $A(G, S, k)$
Input: A graph $G$, set of dominated vertices $S \subseteq V(G)$, an integer $k$
Output: TRUE if $\gamma_g(G|S) \leq k$, FALSE otherwise
  if $S = V(G)$ then
    return TRUE and STOP
  else if $k = 0$ then
    return FALSE and STOP
  else
    for $v \in V(G)$ do
      if $v$ is a legal move then
        if $A'(G, S \cup N[v], k-1)$ then
          return TRUE and STOP
    return FALSE and STOP
```

We first prove the correctness of the algorithms by induction on k. For $k = 0$, it is clear, because $\gamma_g(G|S) = 0$ or $\gamma_g'(G|S) = 0$ if and only if $S = V(G)$. Assume now that the two algorithms are correct for some $k \geq 0$. The algorithm $A(G, S, k+1)$ returns TRUE in two cases. In the first case when $S = V(G)$, we clearly get $\gamma_g(G|S) = 0 < k+1$. In the second case there exists a legal move $v \in V(G)$, such that $A'(G, S \cup N[v], k)$ returns TRUE. By induction hypothesis it follows that $\gamma_g'(G|(S \cup N[v])) \leq k$. Since $\gamma_g(G|S) \leq \gamma_g'(G|(S \cup N[v]))+1$ holds by definition, we derive that $\gamma_g(G|S) \leq k+1$. Conversely, if $A(G, S, k+1)$ returns FALSE, then for any legal move $v \in V(G)$, the algorithm $A'(G, S \cup N[v], k)$

Algorithm $A'(G,S,k)$

```
Input: A graph $G$, set of dominated vertices $S \subseteq V(G)$, an integer $k$
Output: TRUE if $\gamma_g'(G|S) \le k$, FALSE otherwise
  if $S = V(G)$ then
    return TRUE and STOP
  else if $k = 0$ then
    return FALSE and STOP
  else
    for $v \in V(G)$ do
      if $v$ is a legal move then
        if not $A(G, S \cup N[v], k-1)$ then
          return FALSE and STOP
    return TRUE and STOP
```

returns FALSE. By induction hypothesis $\gamma_g'(G|(S \cup N[v])) > k$ holds for all legal moves $v \in V(G)$. That proves that $\gamma_g(G|S) > k+1$. In conclusion the algorithm $A(\cdot,\cdot,k+1)$ is correct. In a similar way, we prove the correctness of $A'(\cdot,\cdot,k+1)$.

Now we prove that the algorithms run in the announced time complexity. The data structure we use for a partially dominated graph $G|S$ is the adjacency list for vertex-weighted graphs, where a vertex of G has weight 1 if it still needs to be dominated, and weight 0 otherwise. That is, vertices from S receive weight 0.

We show by induction on k that for a given graph G of order n, Algorithms A$(\cdot,\cdot,k)$ and A$'(\cdot,\cdot,k)$ both run in $\mathcal{O}(\Delta n^k)$ time. If $k = 1$, then for A we have to check at most n times if a move v is legal and if $N[v] \cup S = V(G)$. For this sake we first compute S and store $|S|$ which is done in time $\mathcal{O}(n)$. After that when going in the loop for vertex v, we compute $|N[v] \setminus S|$. It takes time at most $\mathcal{O}(\Delta)$. Since v is a legal move if and only if $|N[v] \setminus S| > 0$, computing if this move is legal and if $N[v] \cup S = V(G)$ can be done in constant time. In conclusion, we need time at most $\mathcal{O}(n\Delta)$. The same conclusion holds for Algorithm A$'$.

If $k > 0$ then, for Algorithm A, we have to check at most n times whether the move $v \in V(G)$ is legal. We have already seen that this can be implemented in $\mathcal{O}(n\Delta)$ time. Also, the algorithm A$'(G, S \cup N[v], k-1)$ must be run at most n times. We need $\mathcal{O}(n)$ time to build $G|(S \cup N[v])$ and by induction hypothesis, $\mathcal{O}(\Delta n^{k-1})$ time to run A$'(G, S \cup N[v], k-1)$. In conclusion, the time needed is $\mathcal{O}(n\Delta + n \cdot (n + \mathcal{O}(\Delta n^{k-1}))) = \mathcal{O}(\Delta n^k)$.

By a parallel argument we obtain the same complexity for Algorithm A$'$. □

In a practical implementation it would be more efficient to use in Algorithm A the recursive call $\mathrm{A}'(G-v, S\cup N(v), k-1)$. However, this modification does not improve the theoretical complexity of the algorithm.

3. Faster Algorithm for Graphs with $\gamma_g = 3$ and $\mathrm{diam} = 6$

In this section we show that in particular cases the complexity of Algorithms A and A$'$ can be improved. To be more specific, consider the class of graphs E_3^6 defined as follows:

$$E_3^6 = \{G:\ \gamma_g(G) = 3, \mathrm{diam}(G) = 6\}\,.$$

Here the diameter is not selected randomly, the reason to select $\mathrm{diam}(G) = 6$ is that it is the largest possible diameter a graph G with $\gamma_g(G) = 3$ can have. In the recent paper [13], graphs from E_3^6 have been characterized and we are going to use this characterization for an algorithm faster than the canonical one. For this sake, we need to recall the following concept(s).

If G is a connected graph, then a vertex u of G is called *nice* if the following five conditions are fulfilled.

(1) There exists $v_1 \in S_1(u)$ such that $N[v_1] = B_2(u)$.
(2) There is a join between $N(S_3(u)) \cap S_2(u)$ and $S_3(u)$, a join between $S_3(u)$ and $S_4(u)$, and a join between $S_5(u)$ and $S_6(u)$.
(3) The spheres $S_3(u)$ and $S_6(u)$ induce cliques.
(4) There exists $v_5 \in S_5(u)$ such that $S_4(u) \cup S_5(u) \subseteq N[v_5]$.
(5) For any vertex $x \in S_4(u)$ (resp. $S_5(u)$), there exists a vertex $x' \in S_5(u) \cup S_6(u)$ (resp. $S_3(u) \cup S_4(u)$) such that $S_4(u) \cup S_5(u) \subseteq N[x] \cup N[x']$.

Using the concept of a nice vertex, the above mentioned characterization from [13] reads as follows.

Theorem 3.1. *If G is a connected graph, then the following statements are equivalent.*

(i) The graph G belongs to E_3^6.
(ii) Any diametrical pair of vertices contains at least one nice vertex.
(iii) There exists a nice diametrical vertex.

As already mentioned, applying the canonical algorithm from Section 2, graphs G from E_3^6 can be recognized in time $\mathcal{O}(\Delta \cdot |V(G)|^3)$. Using the above theorem, we can substantially improve this complexity as the next result asserts.

Theorem 3.2. *Deciding whether a given graph G belongs to E_3^6 can be implemented in time $\mathcal{O}(|V(G)| \cdot |E(G)| + \Delta^3)$.*

Proof. We will prove that the following algorithm recognizes graphs in E_3^6 within the claimed time complexity.

Algorithm Rec-$E_3^6(G)$

```
Input: A (connected) graph $G$
Output: TRUE if $G \in E_3^6$, FALSE otherwise
  for $v \in V(G)$ do
    determine $S_0(v), \ldots, S_{\mathrm{ecc}(v)}(v)$ and $s_i(v) = |S_i(v)|$, $0 \le i \le \mathrm{ecc}(v)$
  if not $\mathrm{diam}(G) = 6$ then
    return FALSE
  else
    select a vertex $u$ with $\mathrm{ecc}(u) = 6$ and a vertex $u'$ from $S_6(u)$
    if $u$ is nice or $u'$ is nice then
      return TRUE
    else
      return FALSE
```

We first claim that Algorithm Rec-$E_3^6(G)$ is correct. If it returns TRUE, then G has diameter 6 and contains a nice vertex, hence by Theorem 3.1(iii) G belongs to E_3^6. On the other hand, when Rec-$E_3^6(G)$ returns FALSE, there are two possibilities. First, $\mathrm{diam}(G) \neq 6$, which obviously implies that G is not in E_3^6. Second, there exists one pair of diametrical vertices such that both of them are not nice and by Theorem 3.1(ii) we then infer that G does not belong to E_3^6. This proves the correctness of the algorithm.

We next consider the complexity of Algorithm Rec-$E_3^6(G)$. To simplify the notation set $n = |V(G)|$ and $m = |E(G)|$. Using the standard BFS, the spheres around each vertex can be determined in time $\mathcal{O}(m)$. Hence, the first loop of the algorithm can be performed in $\mathcal{O}(nm)$ time. If $\mathrm{diam}(G) = 6$, then the algorithm checks if one of the vertices from a selected diametrical pair u, u' is nice. The corresponding conditions can be verified sequentially. To further simplify the notation we will write S_i to refer to either $S_i(u)$ or to $S_i(u')$. The five conditions can be then verified as follows.

Condition 1: For any vertex $v \in S_1$, $N[v] = S_0 \cup S_1 \cup S_2$ if and only if $\deg(v) = s_0 + s_1 + s_2$. Hence, Condition 1 can be verified in time $\mathcal{O}(s_1\Delta)$.

Condition 2: There is a join between $N(S_3) \cap S_2$ and S_3 if and only if, for any vertex $v \in N(S_3) \cap S_2$, $|N(v) \cap S_3| = s_3$. Hence, we can check that there is such a join in time $\mathcal{O}(s_2\Delta)$. In the same way, we can verify the two other join conditions. All in all, checking Condition 2 can be done in $\mathcal{O}((s_2 + s_3 + s_6)\Delta)$.

Condition 3: This condition needs time $\mathcal{O}((s_3 + s_6)\Delta)$.

Condition 4: This condition needs time $\mathcal{O}(s_5\Delta)$.

Condition 5: For each vertex in S_4 (resp. S_5), we have to find one vertex in $S_5 \cup S_6$ (resp. $S_3 \cup S_4$) which fulfils the given condition. We get that the required time is $\mathcal{O}(s_4(s_5+s_6)\Delta + s_5(s_3+s_4)\Delta)$.

By the above it follows that the first four conditions can be verified in time $\mathcal{O}(nm)$. For the last condition we have to be a bit more careful. Indeed, if we would bound s_3, s_4, s_5, and s_6 above by n, we would get the complexity $\mathcal{O}(n^2\Delta)$. However, note that Condition 5 is tested only if Condition 2 has been successfully tested before. This ensures that s_3, s_4, s_5, and s_6 are bounded above by Δ. Therefore, the complexity of testing Condition 5 is $\mathcal{O}(\Delta^3)$. We conclude that the complexity of Algorithm Rec-$E_3^6(G)$ is $\mathcal{O}(nm + \Delta^3)$. $\square$

Acknowledgements

The research was in part financed by Slovenian Research Agency under the grants P1-0297 and L7-5554, by the ANR-14-CE25-0006 project of the French National Research Agency, and by the grant CMIRA Explo'RA Doc from La Région Rhône Alpes.

References

1. B. Brešar, P. Dorbec, S. Klavžar, G. Košmrlj, *Complexity of the Game Domination Problem*, manuscript, 2014.
2. B. Brešar, S. Klavžar, G. Košmrlj, D. F. Rall, Domination Game: Extremal Families of Graphs for the 3/5-Conjectures, *Discrete Appl. Math.*, **161**, (2013), 1308–1316.
3. B. Brešar, S. Klavžar, G. Košmrlj, D. F. Rall, Guarded subgraphs and the Domination Game, *Discrete Math. Theor. Comput. Sci.*, **17**, (2015), 161–168.
4. B. Brešar, S. Klavžar, D. F. Rall, Domination Game and an Imagination Strategy, *SIAM J. Discrete Math.*, **24**, (2010), 979–991.
5. Cs. Bujtás, Domination Game on Forests, *Discrete Math.*, **38**, (2015), 2220–2228.
6. Cs. Bujtás, On the Game Domination Number of Graphs with Given Minimum Degree, *Electron. J. Combin.*, **22**, (2015), P3.29.
7. Cs. Bujtás, Zs. Tuza, The Disjoint Domination Game, *Discrete Math.*, in press, DOI:10.1016/j.disc.2015.04.028..
8. P. Dorbec, G. Košmrlj, G. Renault, The Domination Game Played on Unions of Graphs, *Discrete Math.*, **338**, (2015), 71–79.
9. M. A. Henning, W. B. Kinnersley, Bounds on the Game Domination Number, manuscript, 2014.
10. M. A. Henning, S. Klavžar, D. F. Rall, Total Version of the Domination Game, *Graphs Combin.*, **31**, (2015), 1453-1462.
11. M. A. Henning, S. Klavžar, D. F. Rall, The 4/5 Upper Bound on the Game Total Domination Number, *Combinatorica*, to appear.
12. W. B. Kinnersley, D. B. West, R. Zamani, Extremal Problems for Game Domination Number, *SIAM J. Discrete Math.*, **27**, (2013), 2090–2107.
13. S. Klavžar, G. Košmrlj, S. Schmidt, On Graphs with Small Game Domination Number, manuscript, 2015.
14. G. Košmrlj, Realizations of the Game Domination Number, *J. Comb. Optim.*, **28**, (2014), 447–461.

15. M. J. Nadjafi-Arani, M. Siggers, H. Soltani, Characterisation of Forests with Trivial Game Domination Numbers, *J. Comb. Optim.*, in press, DOI 10.1007/s10878-015-9903-9.