

THE DOMINATION OVER TIME AND ITS DISCRETISATION

NAZANIN ABBASNEZHAD, JAVAD MEHRI-TAKMEH, JAVAD VAKILI*

Department of Applied Mathematics, School of Mathematics Science,
University of Tabriz, 29 Bahman Blvd., Tabriz, Iran

Domination in graphs is well known and has been an extensively researched branch of graph theory. Since the variation over time is one of the important properties of real-world networks, we study the influence of time on the domination problem. In this paper, we introduce the domination over time problem, including time delay on arcs. Then, an optimal solution to its discretisation is obtained, which is the solution of the original problem.

Keywords: *domination over time, continuous linear programming, duality, discretisation*

1. Introduction

Domination in graphs has been an extensively researched branch of graph theory with applications in various fields, such as computer communication networks, land surveying, monitoring communication and facility locations, like the location of hospital and fire station [12, 22]. Domination set S is a subset of the vertices in a graph such that every vertex in the graph either belongs to S or has a neighbour in S . The standard domination problem searches a domination set of minimum cardinality. The first results in domination were published by Cockayne and Hedetniemi [7]. After that, domination in graphs has become an active field of research [5, 12]. Several polynomial time algorithms have been constructed on trees, interval graphs, and permutation graphs for the domination problem. Cockayne et al. [6] gave the first linear-time algorithms for the domination problem in trees.

The parameter variation over time is a crucial characteristic of networks in real-world applications. This parameter is studied on network flow problems. Ford and Fulkerson [9, 10] introduced the maximum flow over time and then several researchers studied the problem [8, 23]. This paper aims to study the influence of time on the domination problem.

*Corresponding author, email address j.vakili@tabrizu.ac.ir

Received 5 May 2019, accepted 27 April 2020

First, some notations and the programming models, which are needed later, are introduced. Let $G=(V, E)$ be a simple graph with a vertex set V of cardinality $|V|=n$ and an arc set E . Each arc (i, j) is with a time delay (transit time of one unit of data) τ_{ij} .

The adjacency matrix of the graph G is denoted by A .

Hedetniemi et al. [13] introduced the concept of fractional domination problem in graphs. This problem can be formulated as a linear programming problem; hence, it can be solved in polynomial time.

Definition 1.1. A fractional dominating function on a graph G is a function $f:V \rightarrow [0, 1]$ such that

$$f(j) + \sum_{i \in V} A_{ij} f(i) \geq 1, \quad \forall j \in V$$

A minimum fractional dominating function on a graph G is a fractional dominating function f with the minimum value of $|f| = \sum_{j \in V} f(j)$. This minimum value is denoted by $\gamma_f(G)$ and called the fractional domination number of G .

1.1. Separated continuous linear program (SCLP)

In an attempt to model the domination over time problem, we need to present the separated continuous linear program (SCLP). Bellman [4] first introduced the continuous linear program (CLP) to model some economic processes. The SCLP is a subclass of CLP introduced first by Anderson [1] to model job-shop scheduling problems. This problem was studied in the class of continuous linear programs [3, 14, 24]. In particular, Pullan [16, 21] developed an algorithm to solve SCLP under the assumption that the input functions are piecewise linear and piecewise constant. He also studied SCLP with piecewise analytic input functions and developed a duality theory with some conditions under which an optimum extreme point solution exists with a finite number of breakpoints [17–19]. The separated continuous linear program (SCLP) is in the following form:

$$\text{SCLP: } \min \int_0^T c(t)'x(t)dt$$

s.t.

$$\int_0^t Kx(s)ds + y(t) = a(t), \quad t \in [0, T]$$

$$Hx(t) \leq b(t), \quad t \in [0, T]$$

$$x(t), y(t) \geq 0, \quad t \in [0, T]$$

where $c(t)$, $a(t)$, $b(t)$ are vector-valued functions defined on the time interval $[0, T]$, of dimensions n_1, n_2, n_3 , respectively, and K and H are fixed-matrices of dimensions $n_2 \times n_1$ and $n_3 \times n_1$, respectively. The decision variables are vector valued functions $x(t)$ and $y(t)$ of dimensions n_1 and n_2 . All vectors are in columns and the superscript prim ($'$) in the objective function denote the transpose operation. Moreover, $x(t), c(t), b(t)$ are bounded measurable functions, and $y(t), a(t)$ are absolutely continuous functions. Pullan [20] considered a more general class of SCLP with transit-times on arcs called separated continuous linear programs with time-delays (SCLPTD):

$$\begin{aligned} \text{SCLPTD : } & \min \int_0^T c(t)'x(t)dt \\ \text{s.t.} & \\ & \int_0^t (Kx(s))_i ds + \sum_{j=1}^n \int_0^t f_{ij}x_j(s - \tau_{ij})ds + y_i(t) = a_i(t) \\ & t \in [0, T], \quad i = 1, \dots, n_2 \\ & Hx(t) \leq b(t), \quad t \in [0, T] \\ & x(t), y(t) \geq 0, \quad t \in [0, T] \end{aligned}$$

Pullan assumes that the transit times τ_{ij} and the time horizon T are all rational and input functions $c(t)$, $a(t)$ and $b(t)$ are piecewise analytic. Then, he characterises extreme point solutions for SCLPTD and proves the existence of an optimum extreme point solution which is piecewise analytic. Furthermore, he presents an algorithm for solving SCLPTD that $c(t)$ and $a(t)$ are piecewise linear (with $a(t)$ continuous) and $b(t)$ is piecewise constant.

We introduce the domination over time problem (*DOT*) and then solve it under some assumptions on the problem data to determine the optimal solution. The constant and rational transit times are considered, and we find the solution by using discretisation. That is, we show that the optimal solution of *DOT* problem is obtained by a discrete approximation.

The paper is organised as follows: in Section 2, we introduce the *DOT* problem. The dual problem for the *DOT* called *DOT** is introduced in Section 3. We also present and prove the weak duality results between the *DOT* and *DOT**. Two different discretisations of the *DOT* are constructed by partitioning the time interval $[0, 1]$ into finite number of subintervals, each of which gives a bound (lower and upper) on the optimal value of the *DOT* in Section 4. We show that it is not necessary to solve the two discretised problems for successively finer partitions of time. Then they are used for solving the

DOT that outputs the optimal solution of the discrete approximation in which initial valid partition P is the optimal solution of the *DOT*. In Section 5, we present the results from solving several instances.

2. Domination over time

A generalised model for domination problem, as well as the concepts of functional domination in graphs and functional domination number, are introduced. To illustrate it with an example, consider a multicast data communication network by a graph G in which vertices represent devices and arcs represent direct communication links between pairs of devices. Each device at a vertex u can pass data to the set of directly connected vertices at a variable rate $x_u(t)$ (depending on the time parameter). Assume that there is no error in sending or receiving data and each device has an independent receiver and a sender (it can send and receive data simultaneously) and from time to time we need to check data received or read by all devices in a one-time unit (it may be generalised) to a known time horizon T , simply). This is done by reading and sending data between directly connected devices. Every direct connection has a known delay in sending data but there is no delay in reading data from the storage of each device. We need to identify the sending rate from each device with the minimum total data transferred per time unit in the network. Theoretically, the sending rate is a real-valued function. But, practically, it depends on the underlying systems and machines. Therefore, for a specific problem, we may consider it to be bounded above. On the other hand, it is nonnegative, so it is bounded below.

Definition 2.1. Let $G(V, E)$ be a graph. Suppose that $x_j : \mathbb{R} \rightarrow [0, 1]$, $j \in V$ are bounded measurable functions such that $x_j(t) = 0$, for $t \notin [0, 1]$. A vector $\mathbf{x} = (x_1, \dots, x_j, \dots, x_n)$, $j \in V$ is called a dominating functions vector (DFV) if for every $\theta \in [0, 1]$, we have

$$\int_0^\theta x_j(t) dt + \sum_{i \in V} A_{ij} \int_0^\theta x_i(t - \tau_{ij}) dt \geq \theta, \quad j \in V$$

where A is the adjacency matrix of the graph G and $\tau : E \rightarrow \mathbb{R}_{\geq 0}$ is a (symmetric) delay function. Also, we call such \mathbf{x} feasible trajectories.

Definition 2.2. The functional domination number $\gamma_{fd}(G)$ of a graph G is the minimum of $\sum_{j \in V} \int_0^1 x_j(t) dt$ among all DFVs of G .

The following problem with the time parameter is called the domination over time problem (*DOT*) in which the time variable $t \in [0, 1]$, and $x_j(t)$ is the passing rate of data to every point in time t from node j

$$\begin{aligned}
 \text{DOT: } & \min \sum_{j \in V} \int_0^1 x_j(t) dt \\
 \text{s.t.} & \\
 & \int_0^1 x_j(s) ds + \sum_{i \in V} A_{ij} \int_0^t x_i(s - \tau_{ij}) ds \geq t \\
 & j \in V, \quad t \in [0, 1] \\
 & 0 \leq x_j(t) \leq 1,
 \end{aligned}$$

where A is the adjacency matrix of the graph G and $\tau: E \rightarrow \mathbb{R}_{\geq 0}$ is a (constant symmetric) delay function. The optimal value of the *DOT* is denoted by $\gamma_{\text{dot}}(G)$.

Some definitions and notations are provided, which are used in the sequel.

Definition 2.3. A set $P = \{t_0, t_1, \dots, t_m\}$ is said to be a partition of the time interval $[0, 1]$ if $0 = t_0 \leq t_1 \leq \dots \leq t_m = 1$.

The points in P are called breakpoints.

Definition 2.4. A function f is piecewise constant (linear) concerning the partition $P = \{t_0, t_1, \dots, t_m\}$, if it is constant (linear) on $[t_{k-1}, t_k)$ for $k = 1, \dots, m$. f is piecewise constant (linear) on $[0, 1]$ if it is piecewise constant (linear) for some partition of $[0, 1]$.

Definition 2.5. For a function f we use the notations

$$f(t-) = \lim_{s \rightarrow t^-} f(s), \quad f(t+) = \lim_{s \rightarrow t^+} f(s)$$

when the limits exist.

Definition 2.6. We use the notation $Z[OP]$ to denote the optimum value of an optimisation problem (*OP*). Moreover, the notation $Z[OP, x]$ is used to denote the objective function value of *OP* for a given feasible solution x .

Anderson et al. [3] consider SCLP with conditions presented in the following assumption.

Assumption 2.7. The functions $a(t)$ and $c(t)$ are piecewise linear and the function $b(t)$ is piecewise constant on $[0, T]$.

It is shown that under assumption 2.7, an optimal solution $x(t)$ exists, which is piecewise constant if the feasible region is bounded and nonempty. It is enough to see that assumption 2.7 holds for the *DOT* and the *DOT* has a feasible solution and the upper bound constraints on $x(t)$ implying that the feasible region is bounded.

The following definition is introduced due to the existence of transit times by Hashemi and Nasrabadi [11], and we assume that τ_{ij} is constant and nonnegative.

Definition 2.8. A partition $P = \{t_0, t_1, \dots, t_m\}$, of the time interval, $[0, 1]$ is valid if

- for each arc (i, j) and any breakpoint $t_k \in P$, $t_k + \tau_{ij} \leq 1$, implies $t_k + \tau_{ij} \in P$,
- for each arc (i, j) and any breakpoint $t_k \in P$, $t_k - \tau_{ij} \geq 0$, implies $t_k - \tau_{ij} \in P$.

The set of all valid partitions of $[0, 1]$ is denoted by \mathcal{P} .

Now, we construct a valid partition. Set $P := \{0, 1\}$ and add the required breakpoints as follows: for each node i and each breakpoint $t \in P$, if then, add a breakpoint $t + \tau_{ij}$ to P (if it is not already in P). If $t - \tau_{ji} \geq 0$, $\forall (j, i) \in E$, add a breakpoint $t - \tau_{ji}$ to P (if it is not already in P). We repeat this procedure until no new breakpoints are added to P . This procedure terminates in a finite time because the function τ is piecewise constant.

3. The dual problem

In this section, we present a dual problem for domination over time problem.

$$DOT^* \max: \sum_{j \in V} \int_0^1 t d\pi_j(t) - \eta_j(t) dt$$

s.t.

$$1 + \pi_j(t) + \sum_{i \in V} A_{ij} \pi_i(t + \tau_{ij}) + \eta_j(t) \geq 0 \quad (1)$$

$$j \in V, \quad t \in [0, 1]$$

$$\eta_j(t) \geq 0, \quad j \in V, \quad t \in [0, 1]$$

$\pi_j(t)$ monotonic increasing and right continuous on $[0, 1]$ with $\pi_j(1) = 0$, $j \in V$.

Pullan [16] and Anderson [2] represent the dual of SCLP. Here, the form of the dual of domination over time problem is based on the one that was represented by them. A weak duality result is provided as follows.

Lemma 3.1. (Weak duality). $Z[DOT^*] \leq Z[DOT]$.

Proof. Suppose $x(t)$ and $(\pi(t), \eta(t))$ are feasible for the DOT and DOT^* , respectively and

$$K_j(t) = \int_0^t x_j(s) ds + \sum_i A_{ij} \int_0^t x_i(s - \tau_{ij}) ds - t, \quad z_j(t) = 1 - x_j(t)$$

We have

$$\begin{aligned} & \sum_j \int_0^1 x_j(t) dt - \sum_j \int_0^1 t d\pi_j(t) + \sum_j \int_0^1 \eta_j(t) dt = \sum_j \int_0^1 x_j(t) dt \\ & - \sum_j \int_0^1 \left(\int_0^t x_j(s) ds + \sum_i A_{ij} \int_0^t x_i(s - \tau_{ij}) ds \right) dt - K_j(t) d\pi_j(t) \\ & + \sum_j \int_0^1 (z_j(t) + x_j(t)) \eta_j(t) dt \\ & = \sum_j \int_0^1 x_j(t) dt + \sum_j \int_0^1 \pi_j(t) x_j(t) dt + \sum_j \int_0^1 \sum_i A_{ij} \pi_j(t) x_i(t - \tau_{ij}) dt \\ & + \sum_j \int_0^1 d\pi_j(t) K_j(t) + \sum_j \int_0^1 \eta_j(t) z_j(t) dt + \sum_j \int_0^1 \eta_j(t) x_j(t) dt = \sum_j \int_0^1 x_j(t) dt \\ & + \sum_j \int_0^1 (\pi_j(t) + \eta_j(t)) x_j(t) dt + \sum_j \sum_i A_{ij} \int_0^1 \pi_i(t + \tau_{ij}) x_j(t) dt \\ & + \sum_j \int_0^1 d\pi_j(t) K_j(t) + \sum_j \int_0^1 \eta_j(t) z_j(t) dt \\ & = \sum_j \int_0^1 \left(1 + \pi_j(t) + \sum_i A_{ij} \pi_i(t + \tau_{ij}) + \eta_j(t) \right) x_j(t) dt \\ & + \sum_j \int_0^1 d\pi_j(t) K_j(t) + \sum_j \int_0^1 \eta_j(t) z_j(t) dt \geq 0 \end{aligned}$$

4. Discretisation

In this section, two discretisations are introduced for the problem DP , which are the standard one, and AP . It must be noted that the optimal value of DP provides an upper

bound for the optimal value of the *DOT* problem and the optimal value of AP provides a lower bound for it.

Given a partition $P = \{t_0, t_1, \dots, t_m\} \in \mathcal{P}$, we state the first discretisation, which is $DP(P)$ as follows:

$$DP(P): \min \sum_{j \in V} \sum_{k=1}^m (t_k - t_{k-1}) \hat{x}_j(t_{k-1}+)$$

s.t.

$$\hat{x}_j(t_0+) + \sum_{\substack{i \in V \\ \tau_{ij}=0}} A_{ij} \hat{x}_i(t_0+) - \frac{\hat{K}_j(t_1) - \hat{K}_j(t_0)}{t_1 - t_0} = 1, \quad j \in V$$

$$\hat{x}_j(t_{k-1}+) + \sum_{\substack{i \in V \\ \tau_{ij}=t_{k-1}-\tau_{ij}}} A_{ij} \hat{x}_i(t_{k-1}+) - \frac{\hat{K}_j(t_k) - \hat{K}_j(t_{k-1})}{t_k - t_{k-1}} = 1$$

$$j \in V, \quad k = 2, \dots, m$$

$$0 \leq \hat{x}_j(t_{k-1}+) \leq 1, \quad k = 1, \dots, m$$

$$\hat{K}_j(t_k) \geq 0, \quad k = 1, \dots, m$$

This program is in a time-expanded graph in which each layer corresponds to an interval of the partition P . Let $x_j(t_{k-1})$ denote the sending rate of data at a node j in the time point t_{k-1} in the time-expanded graph. Then, we can determine the sending rate of data x_j at a node j the interval $[t_{k-1}, t_k)$ in G . Since any feasible solution to the *DOT* comprises piecewise constant functions, any feasible solution to $DP(P)$ can be constructed from a feasible solution to the *DOT* with the same objective function value. Formally, we set

$$x(t) = \begin{cases} \hat{x}(t_{k-1}+), & t \in [t_{k-1}, t_k), t = T, k = 1, \dots, m \\ \hat{x}(t_{m-1}+), & \end{cases}$$

$$K(t) = \left(\frac{t_k - t}{t_k - t_{k-1}} \right) \hat{K}(t_{k-1}) + \left(\frac{t - t_{k-1}}{t_k - t_{k-1}} \right) \hat{K}(t_k), \quad t \in [t_{k-1}, t_k], \quad k = 1, \dots, m$$

$K(t)$ and $x(t)$ are called the piecewise linear extension of $\hat{K}(t)$ and the piecewise constant extension of $\hat{x}(t)$, respectively [16]. We have the following result:

Lemma 4.1. Let P be a partition in \mathcal{P} . $DP(P)$ is feasible if and only if the DOT problem is feasible. Moreover, we have $Z[DOT] \leq Z[DP(P)]$.

Since the set of all breakpoints of an optimum solution is unknown, this result cannot be used to solve the DOT problem; however, we can obtain a lower bound on the optimum value of the DOT problem by introducing another discretisation. To compute a lower bound on the optimum value of the DOT problem, we use the dual problem of the DOT and introduce the corresponding discrete approximation.

Given the partition $P = \{t_0, t_1, \dots, t_m\}$, we now construct another discretisation; $AP(P)$. The constraints of $AP(P)$ are the same as those for $DP(Q)$, where Q includes P with each interval split in half, i.e.,

$$Q = \left\{ t_0, \frac{t_0 + t_1}{2}, t_1, \dots, t_{m-1}, \frac{t_{m-1} + t_m}{2}, t_m \right\}$$

It is obvious that the set of feasible solutions of $AP(P)$ is the same as the set of feasible solutions of $DP(Q)$, and we identify $\hat{x}_j(t_k -)$ in $AP(P)$ with $\hat{x}_j\left(\frac{t_{k-1} + t_k}{2} +\right)$ in $DP(Q)$.

$$AP(P): \min \sum_{j \in V} \sum_{k=1}^m \left(\frac{t_k - t_{k-1}}{2} \right) (\hat{x}_j(t_{k-1} +) + \hat{x}_j(t_k -))$$

s.t.

$$\hat{x}_j(t_0 +) + \sum_{\substack{i \in V \\ \tau_{ij}=0}} A_{ij} \hat{x}_i(t_0 +) - \frac{\hat{K}_j\left(\frac{t_0 + t_1}{2}\right) - \hat{K}_j(t_0)}{\frac{t_1 - t_0}{2}} = 1, \quad j \in V$$

$$\hat{x}_j(t_{k-1} +) + \sum_{\substack{i \in V \\ \tau_{ij}=t_{k-1}}} A_{ij} \hat{x}_i(t_{k-1} +) - \frac{\hat{K}_j\left(\frac{t_k + t_{k-1}}{2}\right) - \hat{K}_j(t_{k-1})}{\frac{t_k - t_{k-1}}{2}} = 1$$

$$j \in V, \quad k = 2, \dots, m$$

$$\hat{x}_j(t_k-) + \sum_{\substack{i \in V \\ t_i = t_k - t_{ij}}} A_{ij} \hat{x}_i(t-) - \frac{\hat{K}_j(t_k) - \hat{K}_j\left(\frac{t_k + t_{k-1}}{2}\right)}{\frac{t_k - t_{k-1}}{2}} = 1$$

$$j \in V, k = 1, \dots, m,$$

$$0 \leq \hat{x}_j(t_{k-1}+) \leq 1, k = 1, \dots, m,$$

$$0 \leq \hat{x}_j(t_k-) \leq 1, k = 1, \dots, m,$$

$$\hat{K}_j(t_k), \hat{K}_j\left(\frac{t_k + t_{k-1}}{2}\right) \geq 0, k = 1, \dots, m$$

As mentioned before, the constraints of $AP(P)$ are the same as those for $DP(Q)$. Thus, by lemma 4.1, we can conclude that $AP(P)$ is feasible if and only if the DOT is feasible. In particular, when \hat{x} is a feasible solution for $AP(P)$, we obtain the feasible solution for the DOT problem by

$$x(t) = \begin{cases} \hat{x}(t_{k-1}+), t \in \left[t_{k-1}, \frac{t_{k-1} + t_k}{2} \right), & k = 1, \dots, m \\ \hat{x}(t_k-), t \in \left[\frac{t_{k-1} + t_k}{2}, t_k \right), & k = 1, \dots, m \\ \hat{x}(t_m-), t = T \end{cases}$$

Hence, we have the following lemma:

Lemma 4.2. Let P be a partition in \mathcal{P} . The DOT is feasible if and only if $AP(P)$ is feasible. Moreover, we have

$$Z[AP(P)] \leq Z[DOT]$$

Proof. We prove the inequality. The dual problem of $AP(P)$ is as follows:

$$AP^*(P): \max \sum_{j \in V} \sum_{k=1}^m \left\{ - \left(\frac{t_k - t_{k-1}}{2} \right) (\hat{\pi}_j(t_{k-1}+) + \hat{\pi}_j(t_k-)) \right. \\ \left. - \left(\frac{t_k - t_{k-1}}{2} \right) (\hat{\eta}_j(t_{k-1}+) + \hat{\eta}_j(t_k-)) \right\}$$

s.t.

$$1 + \hat{\pi}_j(t_{k-1}+) + \sum_{\substack{i \in V \\ t = t_{k-1} + \tau_{ij}}} A_{ij} \hat{\pi}_i(t+) + \hat{\eta}_j(t_{k-1}+) \geq 0, \quad k = 1, \dots, m$$

$$1 + \hat{\pi}_j(t_k-) + \sum_{\substack{i \in V \\ t = t_k + \tau_{ij}}} A_{ij} \hat{\pi}_i(t-) + \hat{\eta}_j(t_k-) \geq 0, \quad k = 1, \dots, m$$

$$\hat{\eta}_j(t_{k-1}+), \hat{\eta}_j(t_k-) \geq 0, \quad k = 1, \dots, m$$

$$\hat{\pi}_j(t_k+) - \hat{\pi}_j(t_k-) \geq 0, \quad k = 1, \dots, m-1$$

$$\hat{\pi}_j(t_k-) - \hat{\pi}_j(t_{k-1}+) \geq 0, \quad k = 1, \dots, m$$

$$\hat{\pi}_j(t_m-) \leq 0$$

Suppose $(\hat{\pi}, \hat{\eta})$ is a feasible solution for $AP^*(P)$. By the constraints of $AP^*(P)$ and by the fact that $\pi(t)$ and $\eta(t)$ are piecewise linear with partition P , (π, η) is a feasible solution for DOT^* . We now check the relationship between the objective function values of DOT^* and $AP^*(P)$:

$$\begin{aligned} \int_0^1 t d\pi_j(t) &= t\pi_j(t) \Big|_0^1 - \int_0^1 \pi_j(t) dt = - \int_0^1 \pi_j(t) dt \\ &= - \sum_{k=1}^m \int_{t_{k-1}}^{t_k} \pi_j(t) dt \\ &= - \sum_{k=1}^m \left(\frac{t_k - t_{k-1}}{2} \right) (\hat{\pi}_j(t_{k-1}+) + \hat{\pi}_j(t_k-)) \end{aligned} \quad (2)$$

Since $\eta(t)$ is piecewise linear, we have

$$- \int_0^1 \eta_j(t) dt = - \sum_{k=1}^m \left(\frac{t_k - t_{k-1}}{2} \right) (\hat{\eta}_j(t_{k-1}+) + \hat{\eta}_j(t_k-)) \quad (3)$$

By combining (2) and (3), it can be observed that the values of the objective function of the two problems $AP^*(P)$ for $(\hat{\pi}, \hat{\eta})$ and DOT^* for (π, η) are the same. Hence, we obtain:

$$Z[AP^*(P), \hat{\pi}, \hat{\eta}] \leq Z[DOT^*, \pi, \eta] \quad (4)$$

By Lemma 3.1, (4) and the duality theorem for finite dimensional linear programming ($Z[AP(P)] = Z[AP^*(P)]$), we have $Z[AP(P)] \leq Z[DOT]$.

The relationship between the values of various discretisations can be summarised as follows:

Corollary 4.3. For any two partitions P and P' in \mathcal{P} , we have

$$Z[AP(P)] = Z[AP^*(P)] \leq Z[DOT^*] \leq Z[DOT] \leq Z[DP(P')]$$

Proof. The result follows by Lemma 4.2 and lemma 4.1.

Corollary 4.4. For any partition P and P' , if

$$Z[AP(P)] = Z[DP(P')]$$

then the optimal solution of $DP(P')$ gives the optimal solution of DOT .

Proof. Since any feasible solution to $DP(P')$ defines a feasible solution for the DOT with the same value. The result follows from Corollary 4.3.

An important property of the finite dimensional linear programming problems is that the objective function value of the problem is equal to the objective function value of the dual problem in optimality. But this is not always satisfied with continuous time linear programming problems [2]. We show that the optimal value of AP and DOT are the same. So, $Z[DOT] = Z[DOT^*]$ can result from it.

Let P be a partition, \hat{x} be a feasible solution of $AP(P)$ and x is a feasible solution of DOT . We can now define

$$\alpha(\hat{x}) = Z[DOT, x] - Z[AP(P), \hat{x}]$$

Note that by Corollary 4.3, $\alpha(\hat{x}) \geq 0$ for all partitions P and if $\alpha(\hat{x}) = 0$, then $Z[AP(P)] = Z[DP(Q)]$. Hence x is an optimal solution by corollary 4.4.

Lemma 4.5. For any partition P , we have

$$\alpha(\hat{x}) = 0$$

where \hat{x} is an optimum solution for $AP(P)$.

Proof.

$$\begin{aligned} \alpha(\hat{x}) &= \sum_{j \in V} \sum_{k=1}^m \left[\int_{t_{k-1}}^{t_k} x_j(t) dt - \left(\frac{t_k - t_{k-1}}{2} \right) (\hat{x}_j(t_{k-1}+) + \hat{x}_j(t_k-)) \right] \\ &= \sum_{j \in V} \sum_{k=1}^m \left[\hat{x}_j(t_{k-1}+) \left(\frac{t_k + t_{k-1}}{2} - t_{k-1} \right) + \hat{x}_j(t_k-) \left(t_k - \frac{t_k + t_{k-1}}{2} \right) \right. \\ &\quad \left. - \left(\frac{t_k - t_{k-1}}{2} \right) (\hat{x}_j(t_{k-1}+) + \hat{x}_j(t_k-)) \right] = 0 \end{aligned}$$

Lemma 4.5 implies that the optimum values of discrete approximations $DP(Q)$ and $AP(P)$ are the same and the following theorem can be the result.

Theorem 4.6. There is no duality gap between DOT and DOT^* , i.e.,

$$Z[DOT] = Z[DOT^*]$$

The uniform discretisation algorithm, the algorithm presented by Pullan [16], and algorithms presented by Philpott et al. [15] terminate or converge to the optimal solution of SCLP and CNP. Also, Hashemi and Nasrabadi [11] apply these algorithms for dynamic networks flows and present two algorithms that compute, or at least converge, to optimum solutions. We obtain the optimal solution of the DOT by discretisation. We show that the optimal solution of the DOT problem is obtained by solving discrete approximation of the DOT with initial valid P .

5. Numerical examples

In this section, two simple instances of the DOT are solved by using a program written in MATLAB. This program considers the partition and produces $AP(P)$ and $DP(Q)$.

Example 5.1. Consider the example shown in Fig. 1 with six nodes and nine arcs.

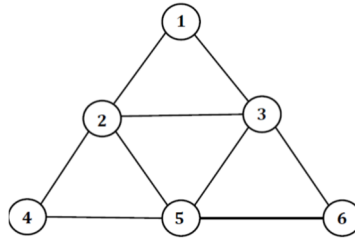


Fig. 1. Hojos graph

The time delays are as follows:

$$\tau_{12} = 0.2, \quad \tau_{13} = 0.5, \quad \tau_{23} = 0.1, \quad \tau_{24} = 0.3, \quad \tau_{25} = 0.4,$$

$$\tau_{35} = 0.7, \tau_{36} = 0.2, \tau_{45} = 0.2, \tau_{56} = 0.1.$$

$P = \{0, 1\}$ is considered and due to definition 2.8, the valid partition is constructed as follows:

$$0 + 0.2 < 1, \quad 0 + 0.5 < 1, \quad 0 + 0.1 < 1,$$

$$0 + 0.3 < 1, \quad 0 + 0.4 < 1, \quad 0 + 0.7 < 1,$$

$$1 - 0.2 = 0.8 > 0, \quad 1 - 0.1 = 0.9 > 0, \quad 1 - 0.4 = 0.6 > 0.$$

So, all of them are added to P . An initial partition $P = \{0, 0.1, 0.2, \dots, 0.9, 1\} \in \mathcal{P}$ is valid. By considering these data, an optimal solution of the *DOT* is obtained as follows:

$$x_1^*(t) = \begin{cases} 1, & t \in [0, 0.2) \\ 0, & t \in [0.2, 1] \end{cases}, \quad x_2^*(t) = \begin{cases} 1, & t \in [0, 0.3) \\ 0, & t \in [0.3, 0.4) \\ 1, & t \in [0.4, 0.45) \\ 0, & t \in [0.45, 1] \end{cases}$$

$$x_3^*(t) = \begin{cases} 1, & t \in [0, 0.35) \\ 0, & t \in [0.35, 0.4) \\ 1, & t \in [0.4, 0.5) \\ 0, & t \in [0.5, 1] \end{cases}, \quad x_4^*(t) = \begin{cases} 1, & t \in [0, 0.2) \\ 0, & t \in [0.2, 1] \end{cases}$$

$$x_5^*(t) = \begin{cases} 1, & t \in [0, 0.1) \\ 0, & t \in [0.1, 0.15) \\ 1, & t \in [0.15, 0.35) \\ 0, & t \in [0.35, 0.4) \\ 1, & t \in [0.4, 0.45) \\ 0, & t \in [0.45, 0.55) \\ 1, & t \in [0.55, 0.6) \\ 0, & t \in [0.6, 0.65) \\ 1, & t \in [0.65, 0.7) \\ 0, & t \in [0.7, 1] \end{cases}, \quad x_6^*(t) = \begin{cases} 1, & t \in [0, 0.1) \\ 0, & t \in [0.1, 1] \end{cases}$$

and $\gamma_{\text{dot}}(G) = 1.75$.

Example 5.2. Consider Fig. 1 with the following time delays:

$$\tau_{12} = 0.2, \quad \tau_{13} = 0.6, \quad \tau_{23} = 0.2, \quad \tau_{24} = 0.4, \quad \tau_{25} = 0.4,$$

$$\tau_{35} = 0.4, \quad \tau_{36} = 0.2, \quad \tau_{45} = 0.2, \quad \tau_{56} = 0.8.$$

A valid partition is $P = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$. An optimal solution is

$$x_1^*(t) = \begin{cases} 1, & t \in [0, 0.2) \\ 0, & t \in [0.2, 1] \end{cases}, \quad x_2^*(t) = \begin{cases} 1, & t \in [0, 0.6) \\ 0, & t \in [0.6, 1] \end{cases}$$

$$x_3^*(t) = \begin{cases} 1, & t \in [0, 0.5) \\ 0, & t \in [0.5, 1] \end{cases}, \quad x_4^*(t) = \begin{cases} 1, & t \in [0, 0.2) \\ 0, & t \in [0.2, 1] \end{cases}$$

$$x_5^*(t) = \begin{cases} 1, & t \in [0, 0.2) \\ 0, & t \in [0.2, 1] \end{cases}, \quad x_6^*(t) = \begin{cases} 1, & t \in [0, 0.2) \\ 0, & t \in [0.2, 0.3) \\ 1, & t \in [0.3, 0.4) \\ 0, & t \in [0.4, 1] \end{cases}$$

and $\gamma_{\text{dot}}(G) = 2$.

As can be seen in the above examples, different time delays on the arcs affect the optimal value. It is easily seen that the optimal solution is not unique in general.

Example 5.3. Consider the example shown in Fig. 2 with 24 nodes. The number next to each arc indicates the delay on the arc.

A valid partition is $P = \{0, 0.1, 0.2, 0.3, \dots, 1\}$. An optimal solution is

$$x_1^*(t) = \begin{cases} 1, & t \in [0, 0.3) \\ 0, & t \in [0.3, 1] \end{cases}, \quad x_2^*(t) = \begin{cases} 1, & t \in [0, 0.4) \\ 0, & t \in [0.4, 1] \end{cases}$$

$$x_3^*(t) = \begin{cases} 1, & t \in [0, 0.3) \\ 0, & t \in [0.3, 1] \end{cases}, \quad x_4^*(t) = \begin{cases} 1, & t \in [0, 0.1) \\ 0, & t \in [0.1, 1] \end{cases}$$

$$x_5^*(t) = \begin{cases} 1, & t \in [0, 0.35) \\ 0, & t \in [0.35, 1] \end{cases}, \quad x_6^*(t) = \begin{cases} 1, & t \in [0, 0.15) \\ 0, & t \in [0.15, 0.2) \\ 1, & t \in [0.2, 0.25) \\ 0, & t \in [0.25, 0.3) \\ 1, & t \in [0.3, 0.35) \\ 0, & t \in [0.35, 0.4) \\ 1, & t \in [0.4, 0.45) \\ 0, & t \in [0.45, 1] \end{cases}$$

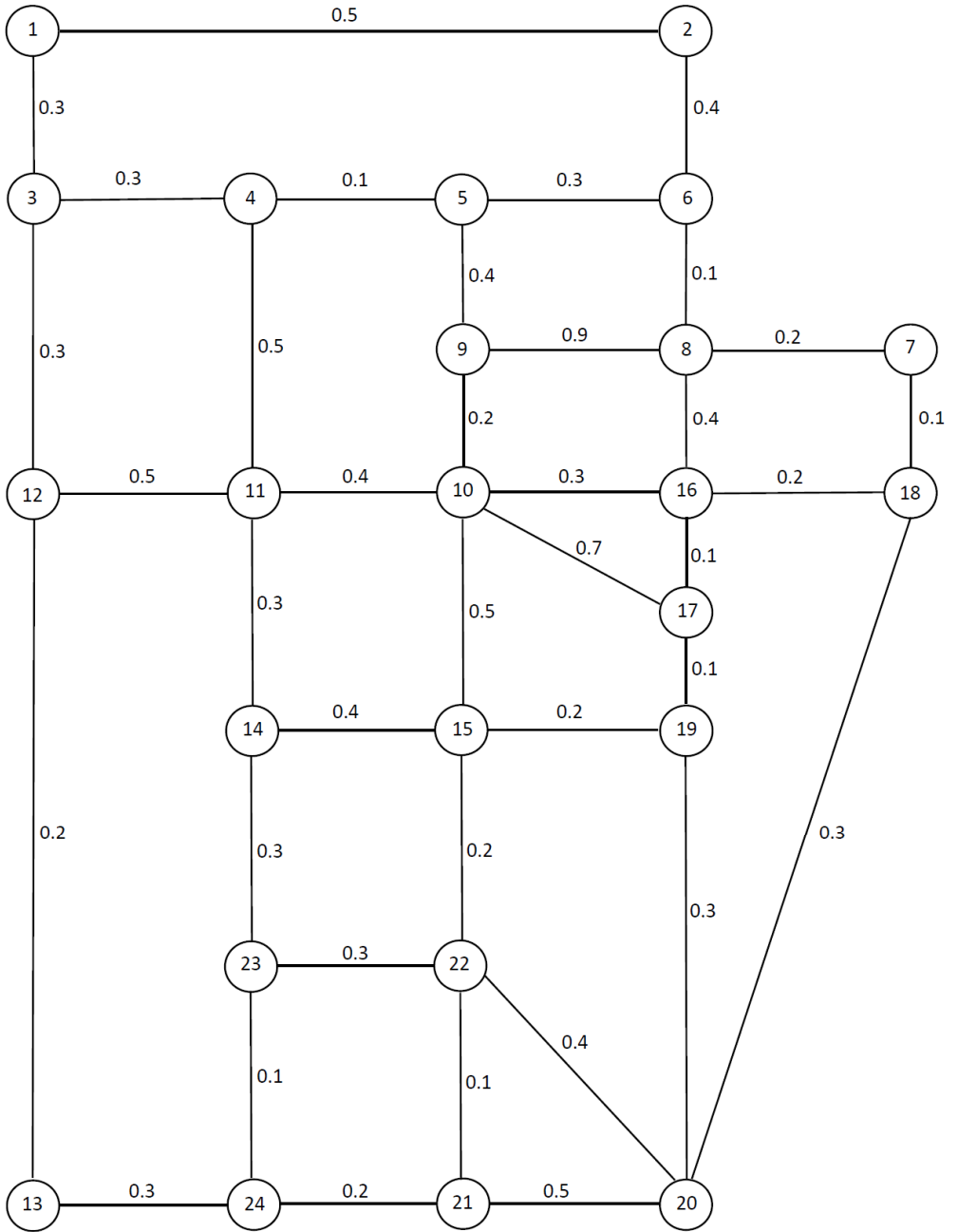


Fig. 2. Sioux falls graph

$$x_7^*(t) = \begin{cases} 1, & t \in [0, 0.1) \\ 0, & t \in [0.1, 0.35) \\ 1, & t \in [0.35, 0.4) \\ 0, & t \in [0.4, 0.45) \\ 1, & t \in [0.45, 0.5) \\ 0, & t \in [0.5, 0.6) \\ 1, & t \in [0.6, 0.75) \\ 0, & t \in [0.75, 0.8) \\ 1, & t \in [0.8, 1] \end{cases}, \quad x_8^*(t) = \begin{cases} 1, & t \in [0, 0.15) \\ 0, & t \in [0.15, 0.35) \\ 1, & t \in [0.35, 0.4) \\ 0, & t \in [0.4, 1] \end{cases}$$

$$x_9^*(t) = \begin{cases} 1, & t \in [0, 0.2) \\ 0, & t \in [0.2, 0.35) \\ 1, & t \in [0.35, 0.4) \\ 0, & t \in [0.4, 1] \end{cases}, \quad x_{10}^*(t) = \begin{cases} 1, & t \in [0, 0.3) \\ 0, & t \in [0.3, 1] \end{cases}$$

$$x_{11}^*(t) = \begin{cases} 1, & t \in [0, 0.3) \\ 0, & t \in [0.3, 1] \end{cases}, \quad x_{12}^*(t) = \begin{cases} 1, & t \in [0, 0.25) \\ 0, & t \in [0.25, 0.55) \\ 1, & t \in [0.55, 0.6) \\ 0, & t \in [0.6, 1] \end{cases}$$

$$x_{13}^*(t) = \begin{cases} 1, & t \in [0, 0.2) \\ 0, & t \in [0.2, 0.6) \\ 1, & t \in [0.6, 0.7) \\ 0, & t \in [0.7, 1] \end{cases}, \quad x_{14}^*(t) = \begin{cases} 1, & t \in [0, 0.3) \\ 0, & t \in [0.3, 1] \end{cases}$$

$$x_{15}^*(t) = \begin{cases} 1, & t \in [0, 0.2) \\ 0, & t \in [0.2, 0.55) \\ 0.5, & t \in [0.55, 0.6) \\ 0, & t \in [0.6, 1] \end{cases}, \quad x_{16}^*(t) = \begin{cases} 1, & t \in [0, 0.1) \\ 0.5, & t \in [0.1, 0.15) \\ 0, & t \in [0.15, 0.25) \\ 1, & t \in [0.25, 0.35) \\ 0, & t \in [0.35, 0.4) \\ 1, & t \in [0.4, 0.45) \\ 0, & t \in [0.45, 1] \end{cases}$$

$$x_{17}^*(t) = \begin{cases} 1, & t \in [0, 0.1) \\ 0, & t \in [0.1, 0.25) \\ 1, & t \in [0.25, 0.3) \\ 0, & t \in [0.3, 0.35) \\ 1, & t \in [0.35, 0.4) \\ 0, & t \in [0.4, 0.5) \\ 1, & t \in [0.5, 0.55) \\ 0, & t \in [0.55, 1] \end{cases}, \quad x_{18}^*(t) = \begin{cases} 1, & t \in [0, 0.1) \\ 0, & t \in [0.1, 0.3) \\ 1, & t \in [0.3, 0.35) \\ 0, & t \in [0.35, 0.4) \\ 1, & t \in [0.4, 0.45) \\ 0, & t \in [0.45, 0.65) \\ 1, & t \in [0.65, 0.7) \\ 0, & t \in [0.7, 1] \end{cases}$$

$$x_{19}^*(t) = \begin{cases} 1, & t \in [0, 0.1) \\ 0, & t \in [0.1, 0.25) \\ 1, & t \in [0.25, 0.3) \\ 0, & t \in [0.3, 0.55) \\ 0.5, & t \in [0.55, 0.6) \\ 0, & t \in [0.6, 1] \end{cases}, \quad x_{20}^*(t) = \begin{cases} 1, & t \in [0, 0.3) \\ 0, & t \in [0.3, 0.45) \\ 1, & t \in [0.45, 0.5) \\ 0, & t \in [0.5, 1] \end{cases}$$

$$x_{21}^*(t) = \begin{cases} 1, & t \in [0, 0.1) \\ 0, & t \in [0.1, 0.25) \\ 0.5, & t \in [0.25, 0.3) \\ 0, & t \in [0.3, 1] \end{cases}, \quad x_{22}^*(t) = \begin{cases} 1, & t \in [0, 0.1) \\ 0, & t \in [0.1, 0.2) \\ 0.5, & t \in [0.2, 0.25) \\ 0, & t \in [0.25, 1] \end{cases}$$

$$x_{23}^*(t) = \begin{cases} 1, & t \in [0, 0.1) \\ 0, & t \in [0.1, 0.4) \\ 1, & t \in [0.4, 0.45) \\ 0, & t \in [0.45, 0.65) \\ 0.5, & t \in [0.65, 0.7) \\ 0, & t \in [0.7, 1] \end{cases}, \quad x_{24}^*(t) = \begin{cases} 1, & t \in [0, 0.2) \\ 0, & t \in [0.2, 0.25) \\ 1, & t \in [0.25, 0.3) \\ 0, & t \in [0.3, 0.4) \\ 1, & t \in [0.4, 0.45) \\ 0, & t \in [0.45, 0.6) \\ 1, & t \in [0.6, 0.7) \\ 0, & t \in [0.7, 1] \end{cases}$$

and $\gamma_{\text{dot}}(G) = 6.6$.

6. Conclusions

While the applications of domination in graphs are not new, to the knowledge of authors, this is the first attempt to study the influence of time on the domination problems. Using the concept of sending rate, we generalised the domination problem and studied the influence of time on the domination problem by considering the time delay on arcs. We introduced the domination over time problem and obtained the optimal solution of the problem.

It would be interesting to study the time parameter and its contribution to the other kinds of domination problems, such as total domination, distance domination, etc.

References

- [1] ANDERSON E., *A continuous model job shop scheduling*, PhD thesis, University of Cambridge, Cambridge 1978.
- [2] ANDERSON E., NASH P., *Linear programming in infinite-dimensional spaces*, Wiley, Chichester, West Sussex, New York 1978.
- [3] ANDERSON E.J., NASH P., PEROLD A.F., *Some properties of a class of continuous linear programs*, SIAM J. Cont. Opt., 1983, 21 (5), 758–765.
- [4] BELLMAN R., *Dynamic Programming*, 1st Ed., Princeton University Press, Princeton 1957.
- [5] BERGE C., *The Theory of Graphs*, Dover Books on Mathematics, Dover 2001.
- [6] COCKAYNE E., GOODMAN S., HEDETNIEMI S., *A linear algorithm for the domination number of a tree*, Inf. Proc. Lett., 1975, 4 (2), 41–44.
- [7] COCKAYNE E., HEDETNIEMI S., *Towards a theory of domination in graphs*, Networks, 1977, 7 (3), 247–261.
- [8] FLEISCHER L., SKUTELLA M., *Minimum cost flows over time without intermediate storage*, [In:] Proc. 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-03), Society of Industrial and Applied Mathematics, ACM, Baltimore 2003 , 66–75.
- [9] FORD L., FULKERSON D., *Constructing maximal dynamic flows from static flows*, Oper. Res., 1958, 6 (3), 419–433.
- [10] FORD L., FULKERSON D., *Flows in Networks*, Princeton University Press, 1962.
- [11] HASHEMI S.M., NASRABADI E., *On solving continuous-time dynamic network flows*, J. Global Opt., 2012, 53 (3), 497–524.
- [12] HAYNES T., HEDETNIEMI S., SLATER P., *Fundamentals of Domination in Graphs*, Chapman and Hall, CRC Pure and Applied Mathematics, Taylor and Francis, 1998.
- [13] HEDETNIEMI S., HEDETNIEMI S., WIMER T., *Linear time resource allocation algorithms for trees*, Technical Report URI, INFORMS, 1987, 14.
- [14] LUO X., BERTSIMAS D., *A new algorithm for state-constrained separated continuous linear programs*, SIAM J. Cont. Opt., 1998, 37 (1), 177–210.
- [15] PHILPOTT A., CRADDOCK M., *An adaptive discretization algorithm for a class of continuous network programs*, Networks, 1995, 26 (1), 1–11.
- [16] PULLAN M., *An algorithm for a class of continuous linear programs*, SIAM J. Cont. Opt., 1993, 31 (6), 1558–1577.
- [17] PULLAN M., *Forms of optimal solutions for separated continuous linear programs*, SIAM J. Cont. Opt., 1995, 33 (6), 1952–1977.

- [18] PULLAN M., *A duality theory for separated continuous linear programs*, SIAM J. Cont. Opt., 1996, 34 (3), 931–965.
- [19] PULLAN M., *Existence and duality theory for separated continuous linear programs*, Math. Model. Syst., 1997, 3 (3), 219–245.
- [20] PULLAN M., *A study of general dynamic network programs with arc time-delays*, SIAM J. Opt., 1997, 7 (4), 889–912.
- [21] PULLAN M., *Convergence of a general class of algorithms for separated continuous linear programs*, SIAM J. Opt., 2000, 10 (3), 722–731.
- [22] SCHÖBEL A., HAMACHER H., LIEBERS A., WAGNER D., *The continuous stop location problem in public transportation networks*, Asia-Pacific J. Oper. Res., 2009, 26 (1), 13–30.
- [23] SKUTELLA M., *An introduction to network flows over time*, [In:] W. Cook, L. Lovász, J. Vygen (Eds.), *Research Trends in Combinatorial Optimization*, Springer, Berlin 2009, 451–482.
- [24] WEISS G., *A simplex based algorithm to solve separated continuous linear programs*, Math. Progr., 2008, 115 (1), 151–198.