Taxed Congestion Games with Failures

Michal Penn*
mpenn@ie.technion.ac.il

Maria Polukarov†
pmasha@tx.technion.ac.il
mp3@ecs.soton.ac.uk

Moshe Tennenholtz‡
moshet@microsoft.com
moshet@ie.technion.ac.il

December 6, 2008

Abstract

In this paper, we introduce and study Taxed Congestion Games with Failures [TCGFs], extending congestion games with failures [CGFs] to consider costly task submission. We define TCGFs, and prove that TCGFs possess a pure strategy equilibrium. Moreover, we provide an efficient algorithm for the computation of such equilibrium. We also provide a specialized, simpler, algorithm for the case in which all resources are identical.

1 Introduction

Models of congestion arise from many real-life situations and have been analyzed by researchers from several fields. Much of the research of congestion settings deals with the model of *congestion games* introduced by Rosenthal [11]. In a congestion game, a player has to choose from a finite set of resources, and the player's payoff depends only on the number of players choosing his resources. Congestion games have been used to model network routing, task allocation, demand for items to be produced, competition among firms for production

*Faculty of Industrial Engineering and Management, Technion, Haifa 32000, Israel

†The work was done when the author was a Ph.D. student at the Technion, Haifa 32000, Israel. Currently the author is a member of Intelligent Agents and Multimedia Group in Department of Electronics and Computer Science at University of Southampton, SO17 1BJ, Southampton, UK.

‡Microsoft Israel R&D Center, 13 Shenkar Street, Herzeliya 46725, Israel; Faculty of Industrial Engineering and Management, Technion, Haifa 32000, Israel

processes, migration of animals between different habitats, etc. [1, 4, 7, 10, 11]. Given the importance of congestion games, various generalizations of the basic model have been considered in the literature. In particular, congestion games have been generalized to *local-effect games* [2], to *player-specific congestion games* [3], to *weighted congestion games* [3], and to *ID-congesiton games* [5]. In these, generalized, models, the player's payoff depends not only on the number of players choosing his resources but also on the number of players choosing neighboring resources or on players' identities.

Although congestion games are central to the computer science and game theory literature, the above models do not take into consideration the possibility that resources may fail to execute their assigned tasks. However, in the computer-science context of congestion settings, where resources are typically represented by machines, computers or communication lines, resources are obviously prone to failures because of high load, breakage, etc. In order to address this issue, we introduced a study of resource failures in congestion games [8]. We presented a model of *Congestion Games with Failure* [CGFs] that allows to refer to the delay experienced by users who select a particular resource as a function of the number of players who select to use it, as well as to the fact that resources may fail with some probability, and as a result, a player may choose a subset of the resources in order to attempt and perform his task. This model however did not deal with the issue of costs incurred by selecting a resource. Needless to say, dealing in the scope of a well defined model with congestion effects, costly submission, and resource failures, is a desired objective. The model we offer in this paper incorporates all these features. We call it *Taxed Congestion Games with Failures* [TCGFs].

In a TCGF there are n agents and m resources. Each agent has a job that he needs to process, for which he can use each of the resources. Each resource might fail with some probability. As a result, the agent may decide to submit his job to several resources, maximizing his probability of success. The processing time of each resource depends on the number of jobs submitted to that resource, and the agent suffers the cost associated with the resource with the fastest processing time he selected. If all the agent's selected resources fail (or he has not selected any resource) then the agent suffers some incompletion costs. What makes each agent's decision highly non-trivial is that each job submission is costly; that is, in addition to the cost incurred due to the delay, there is some fixed payment (which we term "tax"), which is proportional to the number of selected resources. Therefore, there are two central issues that come into play in TCGFs:

1. There is a tradeoff between the probability of successful job completion and the (fixed) costs of submission.
2. While an agent suffers delay costs which are proportional only to the processing time of the fastest resource, submission to many resources by many agents might make the processing time high due to the increase in congestion level.

The above setting, and the associated tradeoffs, capture in a straightforward manner many realistic situations. This is evidently the situation in the manufacturing domain where resources are machines, and in the context of distributed operating systems, but also in the service industry, in which a resource may be a courier required to deliver a message.

Notice that TCGFs extend CGFs, but should not be viewed as a generalization of classical congestion games. This is due to the fact that in both CGFs and TCGFs we care about the delay incurred by the fastest (the least congested) selected resource which did not fail, and not about the congestion on all selected resources. Thus, an agent's payoff function in CGFs and TCGFs uses the minimum, rather than the sum, operator. A model which extends classical congestion games to deal with failures via additive payoffs is presented in [9].

A major question in the context of models of congestion is the existence of a pure strategy equilibrium. The important result of Rosenthal in [11] is that congestion games always possess pure strategy Nash equilibria. Monderer and Shapley [6] introduced the notions of *potential function* and *potential game* and proved that the construction of a potential function is sufficient for showing the existence of a pure strategy equilibrium. The authors [6] observed that Rosenthal [11] proved his theorem on congestion games by constructing a potential function – hence, every congestion game has a potential function. Moreover, they showed that every finite potential game is isomorphic to a congestion game. Therefore, the classes of finite potential games and congestion games coincide. In cases where a potential function does not exist, showing the existence of a pure strategy equilibrium is typically a non-trivial issue. Since even the model introduced in [8] does not admit a potential function, the TCGF-model introduced in this paper (which in particular is a strict generalization of the former model) does not possess a potential function. Hence, the question of whether a pure strategy Nash equilibrium exists in TCGFs, as well as the complexity of finding it, are of considerable significance.

In this paper, we prove the existence of a pure strategy Nash equilibrium in any TCGF, and present a polynomial time algorithm for constructing such an equilibrium. In addition, we develop a simple efficient procedure for computing a pure strategy equilibrium in symmetric TCGFs, where all resources are identical. Our contributions can therefore be summarized as follows:

1. In Section 2 we introduce the first model to capture congestion effects, costly submission, and resources failures, in a game-theoretic setting.
2. In Section 3 we prove the existence of pure strategy equilibria in any TCGF.
3. We show that pure strategy equilibria can be efficiently constructed for TCGFs; this will be implied by the constructive proof introduced in Section 3.
4. We show an efficient algorithm targeted at symmetric TCGFs. This topic is dealt with in Section 4.

2 The model

Our model is an extension of the CGF-model [8]. In a CGF, players share a common set of resources, where each resource may fail with some known probability. Each player has a task which can be carried out by any of the resources. For reliability reasons, a player may choose a subset of the resources in order to try and perform his task. The cost of a player for utilizing any resource is a function of the total number of players using this resource, and the cost for a player for successful completion of his task is the minimum among the costs of his successful attempts. The class of CGFs and, in particular, the subclass of symmetric CGFs, in which the parameters of the game do not depend on the players' or resources' identities, does not admit a potential function; that is, CGFs cannot be reduced to classic congestion games. Nevertheless, as shown in [8], these games always possess a pure strategy Nash equilibrium.

We extend the CGF-model by making task submission costly: in a TCGF, each player pays a fixed cost (tax) for using each of the resources he had chosen, independently of its success or failure. Our extension is accompanied by a limiting assumption of identical failure probabilities. Below we present the formal definition of the TCGF-model.

Let $N = \{1, \dots, n\}$ be a finite set of players, and let $M = \{e_1, \dots, e_m\}$ be a finite set of resources where each resource may fail to execute its assigned tasks with a given *failure probability*. We assume that the failure or success of a particular resource is *independent* of the failure or success of other resources. We also assume that all the resources possess equal failure probabilities. We denote the failure probability of each resource by f $(0 < f < 1)$. Similarly, $s = 1 - f$ stands for the *success probability*.

The set of pure strategies, Σ_i, for player $i \in N$ is the power set of the set of resources, i.e. $\Sigma_i = \mathcal{P}(M)$; the set of pure strategy profiles of all players is denoted by Σ. Player $i \in N$ chooses a strategy $\sigma_i \in \Sigma_i$ which is a (possibly empty) subset of the resources. Player i's disutility from an uncompleted task is evaluated by his *incompletion cost*, w_i. The *service cost* of resource e for each of its users is a nonnegative nondecreasing function $l_e : \{1, \dots, n\} \to \mathbb{R}_+$ of the congestion experienced by e. In addition, there is a (nonnegative) *fixed cost (tax)* (denoted by t) that has to be paid by each player for every resource he uses, independently of its success or failure.

Let $\sigma \in \Sigma$ be a strategy profile; the *congestion vector* that corresponds to σ is $h(\sigma) = (h_e(\sigma))_{e \in M}$, where $h_e(\sigma)$ represents the total number of users of e in σ. The *outcome* from σ is the subset $X \subseteq M$ of the resources that have successfully executed their assigned tasks. For any player $i \in N$, we say that the execution of player i's task *succeeds* if the task of player i is successfully completed by *at least one* of the resources chosen by him, i.e. $\sigma_i \cap X \neq \varnothing$; otherwise, if $\sigma_i \cap X = \varnothing$, player i's task *fails*. The *disutility* for player i from the strategy profile σ and outcome X, is denoted by $\pi_i(\sigma, X)$: player i's disutility from an uncompleted task is evaluated by his incompletion cost, that is $\pi_i(\sigma, X) = w_i$; player i's disutility from a successful completion of his task is determined by the <u>minimum</u> among the service costs of his successful resources:

$$\pi_i(\sigma, X) = \min_{e \in \sigma_i \cap X} l_e(h_e(\sigma)).$$

The *cost* for player i incurred by strategy profile σ and outcome X, $c_i(\sigma, X)$, is the sum of his disutility, $\pi_i(\sigma, X)$ and the fixed costs (taxes) over the resources he has utilized:

$$c_i(\sigma, X) = \pi_i(\sigma, X) + \sum_{e \in \sigma_i} t = \pi_i(\sigma, X) + |\sigma_i| t.$$

Given a strategy profile σ, let $\mathbb{X}(\sigma)$ denote a random variable representing the subset of successful resources; $\mathbb{X}(\sigma)$ is distributed over the power set of the resources, $P(M)$, and its distribution is determined by f. The *expected cost* for player i incurred by strategy profile σ, $C_i(\sigma)$, is therefore:

$$C_i(\sigma) \;\; = \;\; w_i f^{|\sigma_i|} + \sum_{A \in \mathcal{P}(\sigma_i) \setminus \{\varnothing\}} \min_{e \in A} l_e \left(h_e(\sigma)\right) s^{|A|} f^{|\sigma_i \setminus A|} + |\sigma_i| t.$$

For simplicity of exposition, for any subset $A \subseteq M$ of a given set of resources, we further denote by $P(A)$ the set of all *nonempty* subsets of A: $P(A) = \mathcal{P}(A) \setminus \{\varnothing\}$. Then, the expected cost of player i can be written as

$$C_i(\sigma) \;\; = \;\; w_i f^{|\sigma_i|} + \sum_{A \in P(\sigma_i)} \min_{e \in A} l_e \left(h_e(\sigma)\right) s^{|A|} f^{|\sigma_i \setminus A|} + |\sigma_i| t.$$

The aim of each player is to minimize his own expected cost.

We notice that if player i chooses an empty set $\sigma_i = \varnothing$ (does not assign his task to any resource), then his expected cost equals his incompletion cost: $C_i(\varnothing, \sigma_{-i}) = w_i$. We also note that any TCGF with zero taxes is a CGF.

As shown in [8], CGFs and, in particular, player- and resource-symmetric (henceforth, "symmetric") CGFs, are not potential games. Obviously, the subclass of symmetric CGFs is included in the class of (symmetric) TCGFs, and therefore, TCGFs and, in particular, symmetric TCGFs, do not admit a potential function. However, as we show in the next section, they always possess a Nash equilibrium in pure strategies.

3 Pure strategy Nash equilibrium

In this section, we present our main result on TCGFs. We show that these games possess a Nash equilibrium in pure strategies, despite the non-existence of a potential function. Our proof is constructive and yields an $O\left(n^2 m^2 (nm + m \log m)\right)$ time procedure for constructing such an equilibrium in any given TCGF.

We show that TCGFs possess the "single profitable move property", previously defined in [9], implying that a strategy profile which is stable under single moves is a Nash equilibrium. This significantly decreases the size of the strategy set that needs to be examined in order to obtain an equilibrium. Furthermore, we show that TCGFs possess an additional property, "the steady DS-stability property" (to be defined in the sequel), that allows us to develop a monotone[1] iterative algorithm that terminates with a Nash equilibrium strategy profile. We start with the definition of single moves and their intuitive and technical characterizations.

Definition 3.1 *For any strategy profile $\sigma \in \Sigma$ and for any player $i \in N$, adding precisely one resource to his strategy, σ_i, is called an **A-move** of i from σ. Similarly, dropping a single resource is called a **D-move**, and switching one resource with another is called an **S-move**.*

Let $\sigma \in \Sigma$, $i \in N$ and $a \in \sigma_i$. We say that a D-move with a is *profitable* for i if $C_i(\sigma_{-i}, \sigma_i \setminus \{a\}) < C_i(\sigma)$. That is,

$$\begin{aligned} & w_i f^{|\sigma_i|-1} + \sum_{A \in P(\sigma_i \setminus \{a\})} \min_{e \in A} l_e(h_e(\sigma))\, s^{|A|} f^{|\sigma_i|-|A|-1} + (|\sigma_i|-1)\, t \\ & < w_i f^{|\sigma_i|} \quad + \sum_{B \in P(\sigma_i)} \min_{e \in B} l_e(h_e(\sigma))\, s^{|B|} f^{|\sigma_i|-|B|} + |\sigma_i| t, \end{aligned}$$

which is equivalent to

$$\sum_{A \in P(\sigma_i \setminus \{a\})} \min_{e \in A} l_e(h_e(\sigma))\, s^{|A|} f^{|\sigma_i|-|A|-1} - \sum_{B \in P(\sigma_i)} \min_{e \in B} l_e(h_e(\sigma))\, s^{|B|} f^{|\sigma_i|-|B|} + w_i s f^{|\sigma_i|-1} < t. \quad (1)$$

Note that for any pair of sets X, Y, the next equality holds:

$$P(X) = P(X \cap Y) \cup P(X \setminus Y) \cup \left\{ \Omega \cup \Psi \middle| \Omega \in P(X \cap Y), \Psi \in P(X \setminus Y) \right\}. \qquad (2)$$

Using (2), (1) can be rewritten as

$$\begin{aligned} & \sum_{A \in P(\sigma_i \setminus \{a\})} \min_{e \in A} l_e(h_e(\sigma))\, s^{|A|} f^{|\sigma_i|-|A|-1} - \sum_{A \in P(\sigma_i \setminus \{a\})} \min_{e \in A} l_e(h_e(\sigma))\, s^{|A|} f^{|\sigma_i|-|A|} \\ & - \sum_{A \in P(\sigma_i \setminus \{a\})} \min_{e \in A \cup \{a\}} l_e(h_e(\sigma))\, s^{|A|+1} f^{|\sigma_i|-|A|-1} + w_i s f^{|\sigma_i|-1} - l_a(h_a(\sigma))\, s f^{|\sigma_i|-1} < t, \end{aligned}$$

which is equivalent to

$$s f^{|\sigma_i|-1} \left(w_i - l_a(h_a(\sigma)) + \sum_{A \in P(\sigma_i \setminus \{a\})} \left(\min_{e \in A} l_e(h_e(\sigma)) - \min_{e \in A \cup \{a\}} l_e(h_e(\sigma)) \right) s^{|A|} f^{-|A|} \right) < t. \quad (3)$$

Assume a is a successful resource. Then, the left hand side of (3) stands for the difference in the expected disutility (cost) of player i, between his strategies $\sigma_i \setminus \{a\}$ and σ_i. Clearly, if this value is less than t, the fixed cost of using a, then dropping a is a profitable move for

[1] That is, the congestion of each resource does not decrease as the algorithm proceeds

player i. For simplicity of exposition, we use the following notation: for $i \in N$, $a \in M$ and $\sigma \in \Sigma$ let

$$\mathbf{C}_i^a(\sigma) = sf^{|\sigma_i|-1}\left(w_i - l_a\left(h_a(\sigma)\right) + \sum_{A \in P(\sigma_i \setminus \{a\})}\left(\min_{e\in A} l_e\left(h_e(\sigma)\right) - \min_{e \in A\cup\{a\}} l_e\left(h_e(\sigma)\right)\right)s^{|A|}f^{-|A|}\right)$$

denote the *marginal cost saving* by resource a at profile σ for player i.

Thus, (3) can be rewritten as $\mathbf{C}_i^a(\sigma) < t$, meaning that relative to σ, dropping a is profitable for i if (and only if) the fixed cost of using a is greater than its marginal cost saving.

Remark 3.2 *Note that due to the monotonicity of $l_e(\cdot)$ for all $e \in M$, $\mathbf{C}_i^a(\cdot)$ (weakly) decreases with the congestion on resource a and increases with the congestion on each of the other resources. The former follows directly from the monotonicity of $l_a(\cdot)$, and the latter is implied by the monotonicity of $l_e(\cdot)$ for all $e \in \sigma_i \setminus \{a\}$, since the following holds for any $A \in P(\sigma_i \setminus \{a\})$:*

$$\min_{e\in A} l_e\left(h'_e\right) - \min_{e\in A\cup\{a\}} l_e\left(h'_e\right) \geq \min_{e\in A} l_e\left(h_e\right) - \min_{e\in A\cup\{a\}} l_e\left(h_e\right),$$

where h' and h are congestion vectors satisfying $h'_e \geq h_e$ for all $e \in \sigma_i \setminus \{a\}$ and $h'_a = h_a$.

We notice that $\min_{e\in A} l_e\left(h_e(\sigma)\right) \geq \min_{e\in A\cup\{a\}} l_e\left(h_e(\sigma)\right)$ for any set A, implying that the sum in equation (3) is nonnegative. Thus, from (3) we derive

$$sf^{|\sigma_i|-1}\left(w_i - l_a\left(h_a(\sigma)\right)\right) < t,$$

which is equivalent to

$$l_a\left(h_a(\sigma)\right) > w_i - \frac{t}{sf^{|\sigma_i|-1}}. \tag{4}$$

Assume $a \in \arg\max_{e\in\sigma_i} l_e\left(h_e(\sigma)\right)$ and assume that a D-move with a is non-profitable for i (i.e., (3) does not hold). Then, the reverse inequality of (4) is satisfied, since $\min_{e\in A} l_e\left(h_e(\sigma)\right) = \min_{e\in A\cup\{a\}} l_e\left(h_e(\sigma)\right)$ for all $A \in P(\sigma_i \setminus \{a\})$. In addition, since $l_a\left(h_a(\sigma)\right) \geq l_e\left(h_e(\sigma)\right)$ for all $e \in \sigma_i$, the above yields $l_e\left(h_e(\sigma)\right) \leq w_i - \frac{t}{sf^{|\sigma_i|-1}}$ for any $e \in \sigma_i$.

Similar inequalities can be derived for A- and S-moves as follows. An A-move from σ with resource $b \notin \sigma_i$ is profitable for i, i.e. $C_i\left(\sigma_{-i}, \sigma_i \cup \{b\}\right) < C_i(\sigma)$, if and only if the marginal cost saving by b at the resulting profile, $(\sigma_{-i}, \sigma_i \cup \{b\})$, is greater than t, the fixed cost, i.e. $\mathbf{C}_i^b(\sigma_i \cup \{b\}, \sigma_{-i}) > t$. In a similar way, we conclude that an S-move from $a \in \sigma_i$ to $b \notin \sigma_i$ is profitable for i, if and only if the marginal cost saving by switching from a to b is positive, or equivalently, if $l_b\left(h_b(\sigma)+1\right) < l_a\left(h_a(\sigma)\right)$. We summarize the above discussion in Observations 3.3 and 3.4 below.

Observation 3.3 *Let $i \in N$, $a, b, \in M$ and $\sigma \in \Sigma$ satisfying $a \in \sigma_i$ and $b \notin \sigma_i$. Then,*

(1) *A D-move with a is profitable for i if and only if $\mathbf{C}_i^a(\sigma) < t$.*

(2) *An A-move with b is profitable for i if and only if $\mathbf{C}_i^b(\sigma_i \cup \{b\}, \sigma_{-i}) > t$.*

(3) *An S-move from a to b is profitable for i if and only if $l_b(h_b(\sigma)+1) < l_a(h_a(\sigma))$.*

Observation 3.4 *Let $i \in N$, $a, b \in M$ and $\sigma \in \Sigma$ satisfying $a \in \sigma_i$ and $b \notin \sigma_i$. Then,*

(1) (i) *If a D-move with a is profitable for i, then $l_a(h_a(\sigma)) > w_i - \frac{t}{sf^{|\sigma_i|-1}}$.*

(ii) *Assume $a \in \arg\max_{e \in \sigma_i} l_e(h_e(\sigma))$. Then, if a D-move with a is non-profitable for i, then $l_e(h_e(\sigma)) \leq w_i - \frac{t}{sf^{|\sigma_i|-1}}$ for all $e \in \sigma_i$.*[2]

(2) (i) *If an A-move with b is non-profitable for i, then $l_b(h_b(\sigma)+1) \geq w_i - \frac{t}{sf^{|\sigma_i|}}$.*[3]

(ii) *Assume $h_b(\sigma)+1 \geq h_e(\sigma)$ for all $e \in \sigma_i$. Then, the reverse of the above inequality holds, if an A-move with b is profitable for i.*

The following lemma implies that any strategy profile, in which no player wishes unilaterally to apply a single A-, D- or S-move, is a Nash equilibrium. More precisely, we show that if there exists a player who benefits from a unilateral deviation from a given strategy profile, then there exists a single A-, D- or S-move which is profitable for him as well. This property is called the "single profitable move property" [9].

Lemma 3.5 (The single profitable move property) *Given a TCGF, let $\sigma \in \Sigma$ be a strategy profile which is not in equilibrium, and let $i \in N$ be a player for which a profitable deviation from σ is available. Then, there is a profitable single move from σ available to i.*

The idea behind the proof is as follows. Assume on the contrary that σ possesses only non-single-move deviations. Each such deviation can be decomposed into a series of single moves. Consider such a deviation, say σ', with a shortest decomposition. Obviously, inverting any of the single moves is strictly non-profitable with respect to σ' (otherwise, it could have been

[2] It is *strictly* non-profitable if and only if the inequality is strict.

[3] Same as above.

omitted from the original deviation to result a shorter decomposition). This, as we show below, implies the existence of a profitable single move from the original profile, σ. The formal proof follows.

Proof: Let $\sigma \in \Sigma$ be a strategy profile which is not in equilibrium and let $i \in N$ be a player who can benefit from a unilateral deviation from σ. Let $PD_i(\sigma)$ denote the set of all profitable deviations of i from σ, that is

$$PD_i(\sigma) = \{x_i \in \Sigma_i : C_i(\sigma_{-i}, x_i) < C_i(\sigma)\}.$$

For any pair of sets A and B, let $\mu(A, B) = \max\{|A \setminus B|, |B \setminus A|\}$. Clearly, if player i deviates from strategy σ_i to strategy x_i by applying a single A-, D- or S-move, then $\mu(x_i, \sigma_i) = 1$, and vice versa, if $\mu(x_i, \sigma_i) = 1$ then x_i is obtained from σ_i by applying exactly one such move.

Let $y_i \in \arg\min_{x_i \in PD_i(\sigma)} \mu(x_i, \sigma_i)$, and assume $\mu(y_i, \sigma_i) > 1$. Then, the following three inequalities hold for any $a \in \sigma_i$ and $b \notin \sigma_i$:

$$C_i(\sigma_{-i}, \sigma_i \setminus \{a\}) \geq C_i(\sigma); \tag{5}$$

$$C_i(\sigma_{-i}, \sigma_i \cup \{b\}) \geq C_i(\sigma); \tag{6}$$

$$C_i\left(\sigma_{-i}, (\sigma_i \setminus \{a\}) \cup \{b\}\right) \geq C_i(\sigma). \tag{7}$$

We consider separately each of the following three cases: (i) $|\sigma_i \setminus y_i| > |y_i \setminus \sigma_i|$, (ii) $|y_i \setminus \sigma_i| > |\sigma_i \setminus y_i|$, and (iii) $|y_i \setminus \sigma_i| = |\sigma_i \setminus y_i|$.

Case (i): Let $a \in \sigma_i \setminus y_i$, and consider the strategy profile $y_i' = y_i \cup \{a\}$ obtained by inverting the D-move with a from σ. Clearly, $\mu(y_i', \sigma_i) = |\sigma_i \setminus y_i'| < |\sigma_i \setminus y_i| = \mu(y_i, \sigma_i)$. Hence, by the choice of y_i, $C_i(\sigma_{-i}, y_i') > C_i(\sigma_{-i}, y_i)$, implying by Observation 3.4(2),

$$l_a(h_a(\sigma)) > w_i - \frac{t}{sf^{|y_i|}}. \tag{8}$$

Let $\bar{a} \in \arg\max_{e \in \sigma_i} l_e(h_e(\sigma))$. By (5) and Observation 3.4(1),

$$l_{\bar{a}}(h_{\bar{a}}(\sigma)) \leq w_i - \frac{t}{sf^{|\sigma_i|-1}} \quad \Rightarrow \quad l_a(h_a(\sigma)) \leq w_i - \frac{t}{sf^{|\sigma_i|-1}}. \tag{9}$$

Now, $|y_i| \leq |\sigma_i| - 1$ (since $|\sigma_i \setminus y_i| > |y_i \setminus \sigma_i|$) impling that (8) contradicts (9).

Case (ii): Let $b \in \arg\max_{e \in y_i \setminus \sigma_i} l_e(h_e(\sigma) + 1)$. By (6) and Observation 3.4(2) we get

$$l_b(h_b + 1) \geq w_i - \frac{t}{sf^{|\sigma_i|}}. \tag{10}$$

In addition, (7) and Observation 3.3(3) imply $l_b(h_b+1) \geq l_a(h_a)$ for all $a \in \sigma_i$. This, coupled with the choice of b, implies that $b \in \arg\max_{e \in y_i} l_e(h_e(\sigma_{-i}, y_i))$.

Consider the strategy profile $y_i'' = y_i \setminus \{b\}$ obtained by inverting the A-move with b from σ. Clearly, $\mu(y_i'', \sigma_i) = |y_i'' \setminus \sigma_i| < |y_i \setminus \sigma_i| = \mu(y_i, \sigma_i)$. Hence, by the choice of y_i, $C_i(\sigma_{-i}, y_i'') > C_i(\sigma_{-i}, y_i)$, implying by $b \in \arg\max_{e \in y_i} l_e(h_e(\sigma_{-i}, y_i)$ and Observation 3.4(1) that

$$l_b(h_b+1) < w_i - \frac{t}{sf^{|y_i|-1}}. \tag{11}$$

Now, $|\sigma_i| \leq |y_i| - 1$ (since $|y_i \setminus \sigma_i| > |\sigma_i \setminus y_i|$) implies that (11) contradicts (10).

Case (iii): Let $a \in \sigma_i \setminus y_i$ and $b \in y_i \setminus \sigma_i$, and consider $y_i''' = (y_i \setminus \{b\}) \cup \{a\}$. Clearly, $\mu(y_i''', \sigma_i) < \mu(y_i, \sigma_i)$. Hence, by the choice of y_i, $C_i(\sigma_{-i}, y_i''') > C_i(\sigma_{-i}, y_i)$, implying by Observation 3.3(3) that $l_a(h_a) > l_b(h_b+1)$. This, in turn, yields $C_i(\sigma_{-i}, \sigma_i') < C_i(\sigma)$, in contradiction to (7). This completes the proof. □

Based on Lemma 3.5, in order to prove the existence of a pure strategy Nash equilibrium, it suffices to present an A-, D- and S-stable strategy profile, as defined below.

Definition 3.6 *A strategy profile σ is said to be **A-stable** (resp., **D-stable**, **S-stable**) if there are no players with a profitable A- (resp., D-, S-) move from σ.*

We notice that the strategy profile $\sigma^0 = (\varnothing, \ldots, \varnothing)$ is D- and S-stable (henceforth, "DS-stable"), so the subset of DS-stable profiles is not empty. Our goal is to find a DS-stable profile for which no profitable A-move exists, implying this profile is in equilibrium. Using the above, we develop an iterative algorithm having the following properties:

- The input and the output of each iteration of the algorithm is DS-stable.
- The congestion of each resource $e \in M$ does not decrease as the algorithm proceeds.
- The algorithm terminates with a Nash equilibrium strategy profile.

Below we present our TNE-algorithm that finds a pure strategy Nash equilibrium in a given TCGF. Let us start with its brief description. The TNE-algorithm is initialized with $\sigma^0 = (\varnothing, \ldots, \varnothing)$, and each of its iterations begins with a DS-stable strategy profile (see Lemma

3.7 in the sequel). Let σ represent the input of an iteration of the TNE-algorithm, and let h denote its corresponding congestion vector ($h = h(\sigma)$). The algorithm sorts the set of all resources $e \in M$ with $h_e < n$ in a non-decreasing order of $l_e(h_e + 1)$. For each player i, let e^i be the smallest resource (according to the current order) not in σ_i. Observe that if there is a profitable A-move of i from σ then the A-move with e^i is a most profitable move for him. If there are no profitable A-moves for any player, then σ is a Nash equilibrium strategy profile and the algorithm terminates. Otherwise, let $\bar{N}$ denote the set of all players who wish to apply an A-move, and let $e_{\min} = \min\{e^i : i \in \bar{N}\}$. The algorithm selects from $\bar{N}$ a player i_A with $e^{i_A} = e_{\min}$, and adds resource $e_{\min}$ to his strategy. If the resulting strategy profile, σ', is DS-stable, then the algorithm proceeds to the next iteration. Otherwise, if σ' is not DS-stable (see Figure 1), it needs to be stabilized. As proved in Lemma 3.7 (Claim 3.10), the only

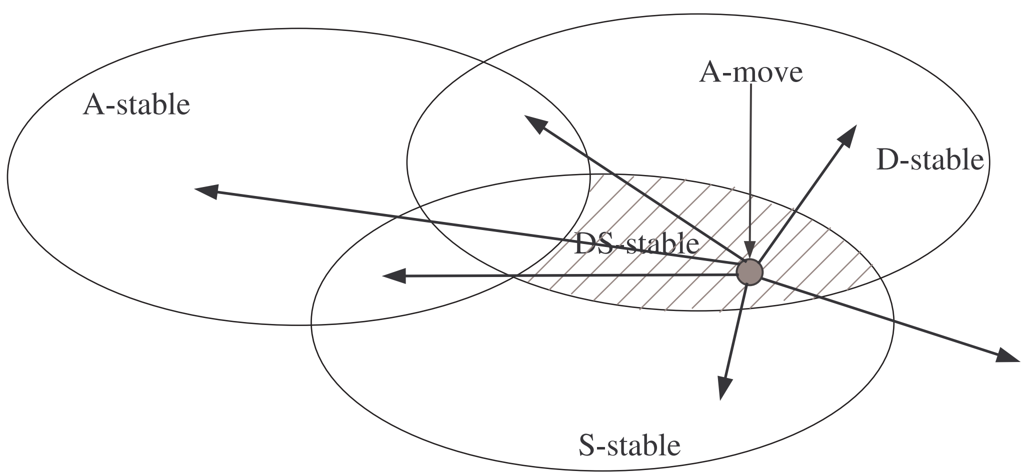

Figure 1: Applying an A-move to a DS-stable profile may destroy the DS-stability.

potential cause for the in-DS-stability of σ' is the existence of a player who wishes to apply a D- or S-move with $e_{\min}$. Let $\widetilde{N}$ denote the set of players who wish to make such a change in their strategies. The algorithm selects player j_D from $\widetilde{N}$, and removes resource $e_{\min}$ from his strategy. Then, Lemma 3.7(Claim 3.11) shows that the resulting strategy profile, σ'', is DS-stable (see Figure 2), and the algorithm proceeds to the next iteration. We note in Remark 3.12 in the sequel, that splitting each S-move into a D-move that is followed by an A-move in the following iteration, does not effect the outcome of the algorithm. Therefore, our procedure considers, in fact, only two kinds of operations: additions and deletions. Furthermore, Lemma 3.8 shows that the algorithm stops with a Nash equilibrium strategy profile, and enables us to derive the time complexity of the algorithm. The TNE-algorithm is presented below.

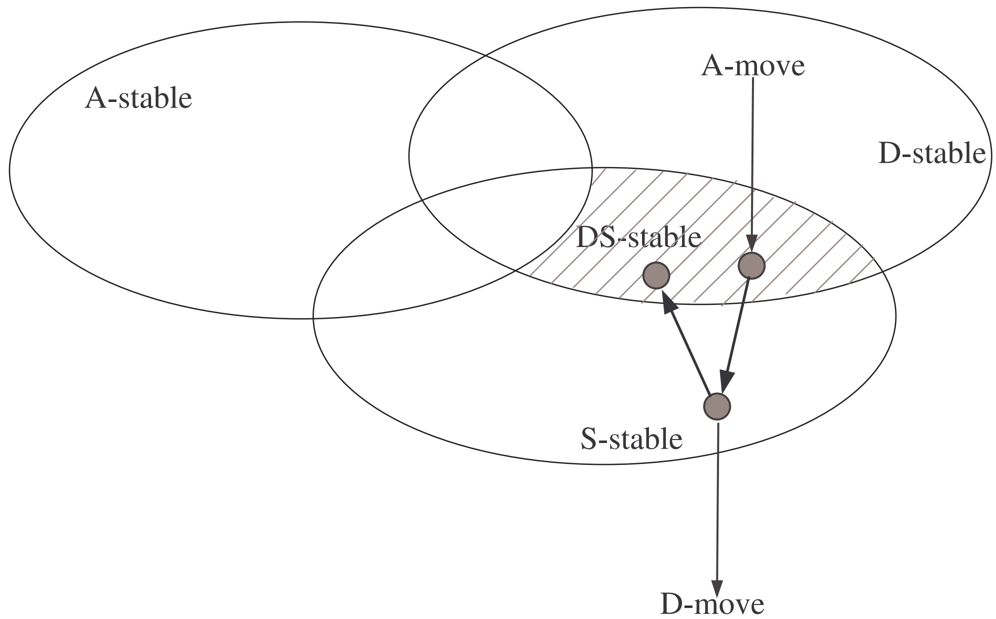

Figure 2: The DS-stability can be fixed by a single D-move.

TNE-algorithm

Initiali-	For all $i \in N$, set $\sigma_i := \varnothing$;
zation:	For all $e \in M$ set $h_e := 0$;
Main	[1] Set $\bar{M} := \{e \in M \,:\, h_e < n\}$;
step:	[2] Reorder $\bar{M}$ according to the rule
	$x \leq y \Leftrightarrow l_x(h_x+1) \leq l_y(h_y+1)$;
	[3] For all $i \in N$, set $e^i := \min\{x \,:\, x \notin \sigma_i\}$;
	[4] If $C_i(\sigma_{-i}, \sigma_i \cup \{e^i\}) \geq C_i(\sigma)$
	for all $i \in N$, then QUIT;
	[5] Set $\bar{N} := \{i \in N \,:\, C_i(\sigma_{-i}, \sigma_i \cup \{e^i\}) < C_i(\sigma)\}$;
	[6] Set $e_{\min} := \min\{e^i \,:\, i \in \bar{N}\}$;
	[7] Select $i_A \in \{i \in \bar{N} \,:\, e^i = e_{\min}\}$;
	$\sigma_{i_A} := \sigma_{i_A} \cup \{e_{\min}\}$;
	$h_{e_{\min}} := h_{e_{\min}} + 1$;
	[8] If $(\sigma_{-i_A}, \sigma_{i_A} \cup \{e_{\min}\})$ is DS-stable,
	then go to (1);

[9] Set $\widetilde{N} := \{j \in N \,:\, C_j(\sigma) > C_j(\sigma_{-j}, \sigma_j \setminus \{e_{\min}\})$ or

$\exists u \in \bar{M} \setminus \sigma_j,\, C_j(\sigma) > C_j(\sigma_{-j}, (\sigma_j \setminus \{e_{\min}\}) \cup \{u\})\}$;

[10] Select $j_D \in \widetilde{N}$;

[11] Set $\sigma_{j_D} := \sigma_{j_D} \setminus \{e_{\min}\}$;

$h_{e_{\min}} := h_{e_{\min}} - 1$, and go to [3].

Lemmas 3.7 and 3.8 below are central for proving the correctness of the TNE-algorithm. For the reason of exposition, we have chosen to present their proofs at the end of this section.

Lemma 3.7 *The output of each iteration of the TNE-algorithm is a DS-stable profile.*

The proof consists of two parts. We start by showing that the only potential cause for an in-DS-stability of a profile obtained by an A-move from a DS-stable profile is the existence of a player, say player j, who used the added resource before the addition operation and wishes to drop it after it had been added by another player (see Claim 3.10). We proceed by showing that if such player j exists, then if he removes the added resource from his strategy then the resulting profile is DS-stable (see Claim 3.11). Thus, Lemma 3.7 implies the existence of a monotone procedure with its inputs and outputs at each iteration being DS-stable strategy profiles.

Clearly, the congestion of each resource does not decrease as the algorithm proceeds. Therefore, in order to prove that the algorithm terminates after finitely many iterations, it suffices to show that every sequence of iterations with a constant congestion is finite. This statement follows from Lemma 3.8, implying that once a player has added a resource to his strategy set, he will not remove it, unless the congestion in the system has been changed.

Lemma 3.8 *Let σ^k represents the input of the k'th iteration of the TNE-algorithm with h^k being its corresponding congestion vector, and let $p = i^k_A$ be a player who adds resource $e^k_{\min}$ to his strategy σ^k_p at the beginning of the k'th iteration. Then, every $e \leq e^k_{\min}$ satisfies $e \in \sigma^r_p$ for all $r > k$ with $h^r = h^k$. That is, player p uses all the resources ordered less than $e^k_{\min}$, as long as the congestion in the system has not been changed.*

We turn now to present our main result.

Theorem 3.9 *The TNE-algorithm finds a pure strategy Nash-equilibrium in a given TCGF in time* $O\left(n^2m^2(nm + m\log m)\right)$.

Proof: *Validity* Let σ be the output of the TNE-algorithm. The algorithm halts if and only if there are no players who wish to unilaterally apply an A-move from σ. That is, σ is an A-stable strategy profile. In addition, by Lemma 3.7, σ is DS-stable. Thus, by Lemma 3.5, σ is a Nash equilibrium strategy profile.

Complexity Each iteration of the algorithm takes $O(nm + m\log m)$ operations for reordering the resources and applying an A- and a D-move. We show below that the number of iterations is bounded by $(nm)^2$. Since the congestion of the resources do not decrease as the algorithm proceeds, the number of possible congestion combinations of the resources (congestion vectors) is bounded by nm. Assume the algorithm starts a new iteration and let h denote the current congestion vector. By Lemma 3.8, a player that adds a resource, at the beginning of the iteration, will not remove it as long as the congestion of the resources remains h. Thus, preserving the same congestion vector, the algorithm can replace the users of each resource at most once. Therefore, the number of iterations with the same congestion vector is bounded by nm, and the complexity of the TNE-algorithm is $O\left(n^2m^2(nm + m\log m)\right)$. □

We proceed with the proofs of Lemmas 3.7 and 3.8.

Proof of Lemma 3.7: The proof is by induction on the iteration depth. For the first iteration the proof is immediate. The input of the first iteration is $\sigma = (\varnothing, \ldots, \varnothing)$ which is obviously DS-stable. If no player wishes to apply an A-move then the output of the first iteration is σ. Otherwise, if there is a player i_A, who wishes to apply an A-move with resource $e_{\min}$, then the output of the first iteration is a strategy profile $\sigma' = (\sigma_{-i_A}, \sigma_{i_A} \cup \{e_{\min}\})$. In this case, $C_{i_A}(\sigma_{-i_A}, \sigma_{i_A} \cup \{e_{\min}\}) < C_{i_A}(\sigma)$, impling that a D-move with $e_{\min}$ from σ' is not profitable for i_A. By the TNE-algorithm, $l_{e_{\min}}(1) \leq l_e(1)$, for all $e \in M$. This yields $C_{i_A}(\sigma_{-i_A}, \sigma_{i_A} \cup \{e_{\min}\}) \leq C_{i_A}(\sigma_{-i_A}, \sigma_{i_A} \cup \{e\})$, for all $e \in M$. That is, i_A does not wish to apply an S-move with $e_{\min}$ from σ'. Since all other players have nothing to drop or exchange, the strategy profile σ' – the output of the first iteration – is DS-stable.

Now we assume that the input of the k'th iteration ($k > 1$) is DS-stable and show that so is its output. Let σ be the DS-stable input of the k'th iteration. If no player wishes to apply an A-move then the output of the k'th iteration is σ. Otherwise, player i_A adds resource

$e_{\min}$ to his strategy. For simplicity of notation, let us denote i_A by i and $e_{\min}$ by a, and let $\sigma' = (\sigma_{-i}, \sigma_i \cup \{a\})$. Then,

Claim 3.10 *Let $j \in N \setminus \{i\}$. Then, the only potential profitable D- or S- move by j from σ' is with a.*

Proof: Let $b \in \sigma'_j \setminus \{a\} = \sigma_j \setminus \{a\}$. By the D-stability of σ and Observation 3.3(1), $\mathbf{C}_j^b(\sigma) \geq t$. Recall that $\sigma' = (\sigma_{-i}, \sigma_i \cup \{a\})$, hence $h_e(\sigma') = h_e(\sigma)$ for all $e \in M \setminus \{a\}$ and $h_a(\sigma') = h_a(\sigma) + 1$, implying $h_e(\sigma') \geq h_e(\sigma)$ for all $e \in M$ and $h_b(\sigma') = h_b(\sigma)$ (since $b \neq a$). Then, by Remark 3.2, $\mathbf{C}_j^b(\sigma') \geq \mathbf{C}_j^b(\sigma) \geq t$, implying by Observation 3.3(1) that a D-move with b is non-profitable for j.

By the S-stability of σ and Observation 3.3(3), for any resource $c \notin \sigma_j = \sigma'_j$ we have $l_c(h_c(\sigma)+1) \geq l_b(h_b(\sigma))$. Now, since $h_e(\sigma') \geq h_e(\sigma)$ for all $e \in M$ and $h_b(\sigma') = h_b(\sigma)$, the above yields $l_c(h_c(\sigma')+1) \geq l_b(h_b(\sigma'))$, implying by Observation 3.3(3) the non-profitability of an S-move from resource b.

Thus, no player $j \neq i$ wishes to apply a D- or S-move from any resource $b \in \sigma'_j = \sigma_j$, $b \neq a$.

$\square_{Claim 3.10}$

If no player $j \neq i$ wishes to apply a D- or S-move with the added resource a, then the output of the k'th iteration is the strategy profile $\sigma' = (\sigma_{-i}, \sigma_i \cup \{a\})$. In this case, we complete the proof by showing that player i also does not wish to apply any D- or S-move. Clearly, after adding resource a, player i does not wish to drop it. Recall that $a \in \arg\min\{l_e(h_e + 1) : e \notin \sigma_i\}$, which yields $C_i(\sigma') \leq C_i(\sigma'_i, (\sigma'_i \setminus \{a\}) \cup \{e\})$ for any $e \notin \sigma'_i$, implying that player i does not wish to exchange resource a by any other resource. It remains to show that i does not wish to drop or exchange any resource $b \in \sigma'_i \setminus \{a\} = \sigma_i$. This follows from the fact that $C_i(\sigma') < C_i(\sigma)$ and the σ's S-stability, which imply the following:

(i) $C_i(\sigma') < C_i(\sigma) \leq C_i(\sigma_{-i}, (\sigma_i \setminus \{b\}) \cup \{a\}) = C_i(\sigma'_{-i}, \sigma'_i \setminus \{b\})$;

(ii) $C_i(\sigma) \leq C_i(\sigma_{-i}, (\sigma_i \setminus \{b\}) \cup \{e\})$, for all $e \notin \sigma_i$. By Observation 3.3(3), this yields $l_b(h_b(\sigma)) \leq l_e(h_e(\sigma) + 1)$ for all $e \notin \sigma_i$ and, in particular, for all $e \notin \sigma'_i$. Now, since $h_e(\sigma') \geq h_e(\sigma)$ for all $e \in M$ and $h_b(\sigma') = h_b(\sigma)$, the above yields $l_b(h_b(\sigma')) \leq l_e(h_e(\sigma')+1)$, implying $C_i(\sigma') \leq C_i(\sigma'_{-i}, (\sigma'_i \setminus \{b\}) \cup \{e\})$, for all $e \notin \sigma'_i$.

If there exists player j_D who wishes to apply a D- or S-move with the added resource a, then the output of the algorithm is a strategy profile $\sigma'' = (\sigma'_{-j_D}, \sigma'_{j_D} \setminus \{a\})$ which is obtained from σ' by the D-move of j_D from a.[4] For simplicity of notation, we denote j_D by j. Then,

Claim 3.11 (The steady DS-stability property) The strategy profile σ'' is DS-stable.

Proof: Note that $\sigma'' = (\sigma_{-\{i,j\}}, \sigma_i \cup \{a\}, \sigma_j \setminus \{a\})$, hence σ'' and σ have the same congestion vector, and all players in $N \setminus \{i,j\}$ do not distinguish between σ'' and σ. Therefore, it remains to show the DS-stability of σ'' with respect to players i and j. For simplicity of exposition, let us denote the congestion vector of σ'' and σ by h ($h = h(\sigma) = h(\sigma'')$).

Consider player i. By the S-stability of σ and Observation 3.3(3),

$$l_e(h_e) \leq l_{e'}(h_{e'} + 1) \tag{12}$$

for all $e \in \sigma_i$ and $e' \notin \sigma_i$, and, in particular, for all $e \in \sigma'' \setminus \{a\}$ and $e' \notin \sigma''_i$. Since $a \in \arg\min\{l_{e'}(h_{e'} + 1) \ : \ e' \notin \sigma_i\}$, then $l_a(h_a + 1) \leq l_{e'}(h_{e'} + 1)$ for all $e' \notin \sigma_i$. By the monotonicity of $l_e(\cdot)$ for all $e \in M$, $l_a(h_a) \leq l_{e'}(h_{e'} + 1)$ for all $e' \notin \sigma_i$, and, in particular, for all $e' \notin \sigma''_i$. Thus, for all $e \in \sigma''$ and $e' \notin \sigma''_i$ we get $l_e(h_e) \leq l_{e'}(h_{e'} + 1)$ and $C_i(\sigma'') \leq C_i(\sigma''_{-i}, (\sigma''_i \setminus \{e\}) \cup \{e'\})$, implying that player i does not wish to apply any S-move from σ''.

By $C_i(\sigma') < C_i(\sigma)$ and the monotonicity of $l_e(\cdot)$ for all $e \in M$,

$$l_a(h_a) \leq l_a(h_a + 1) \ \Rightarrow \ C_i(\sigma'') \leq C_i(\sigma') < C_i(\sigma) = C_i(\sigma''_{-i}, \sigma''_i \setminus \{a\}),$$

implying that player i does not wish to apply a D-move with resource a. We show now that this holds for all other resources in σ''_i.

By (12), $l_a(h_a + 1) \geq l_e(h_e)$ for all $e \in \sigma_i$. Then, By $C_i(\sigma') < C_i(\sigma)$ and Observation 3.4(2),

$$l_a(h_a + 1) < w_i - \frac{t}{sf^{|\sigma_i|}}. \tag{13}$$

Let $b \in \sigma''_i \setminus \{a\} = \sigma_i$. By (13) and $l_a(h_a + 1) \geq l_b(h_b)$ we have $l_b(h_b) < w_i - \frac{t}{sf^{|\sigma_i|}}$, implying $l_b(h_b) < w_i - \frac{t}{sf^{|\sigma''_i| - 1}}$ (since $|\sigma_i| = |\sigma''_i| - 1$). Thus, by Observation 3.4(1), the D-move with b from σ'' is non-profitable for i.

[4] Notice that the algorithm applies a D-move of player j_D from resource a even if j_D would prefer to exchange a by another resource. If this is the case, then at next iteration j_D will be the only player with a profitable A-move and will be selected by the algorithm to apply it (see Remark 3.12 following this proof). That is, the desirable by j_D S-move is just split into two iterations.

Consider now player j. Assume first that a D-move with a from σ' is profitable for j. In this case we derive the DS-stability of σ'' directly from the DS-stability of σ. More precisely, by the D-stability of σ and Observation 3.3(2), for all $e \in \sigma_j$ and, in particular, for all $e \in \sigma''_j$ we get $l_e(h_e) \leq w_j - \frac{t}{sf^{|\sigma_j|-1}} < w_j - \frac{t}{sf^{|\sigma''_j|-1}}$, where the latter inequality follows since $|\sigma''_j| > |\sigma_j|$. Therefore, no profitable D-move from σ'' is available to j.

By the S-stability of σ and Observation 3.3(3), for any $e \in \sigma_j$ and $e' \in M \setminus \sigma_j$ we have $l_e(h_e) \leq l_{e'}(h_{e'} + 1)$, implying that no profitable S-move from σ'' with $e \in \sigma''_j$ and $e' \in (M \setminus \sigma''_j) \setminus \{a\}$ is available to j. By the profitability of the D-move with a from σ' and the D-stability of σ, we get $l_a(h_a+1) > w_i - \frac{t}{sf^{|\sigma_j|-1}}$ and $l_e(h_e) \leq w_i - \frac{t}{sf^{|\sigma_j|-1}}$ for any $e \in \sigma_j$ and, in particular, for any $e \in \sigma''_j$, implying that $l_a(h_a + 1) > l_e(h_e)$. Therefore, we can conclude that no profitable S-move with $e \in \sigma''_j$ and a is available to j, completing the proof of the S-stability of σ'' in this case.

Otherwise, if the D-move with a is not profitable for j, then there is a profitable S-move from a to $c \notin \sigma''_j$, implying that

$$l_c(h_c + 1) < l_a(h_a + 1). \tag{14}$$

As in the previous case, the D-stability of σ'' w.r.t player j follows directly from the D-stability of σ. Let us proceed and prove the S-stability of σ'' w.r.t j. Assume on the contrary that j wants to switch resource $v \in \sigma''_j$ with resource $u \in M \setminus \sigma''_j$. Then, by Observation 3.3(3),

$$l_u(h_u + 1) < l_v(h_v). \tag{15}$$

If $u \neq a$ then (15) contradicts (12). Otherwise, by (14) and (12), $l_a(h_a + 1) > l_v(h_v)$, in contradiction to (15). $\square_{Claim 3.11}$

This completes the proof of the lemma. $\square_{Lemma 3.7}$

Remark 3.12 *Consider the k'th iteration of the algorithm, where an A-move of player i_A^k with resource $e_{\min}^k$ destabilizes the system. If after adding i_A^k to $e_{\min}^k$ player j_D^k prefers to remove $e_{\min}^k$ from his strategy, then he will not wish to add it to his strategy at the next iteration, i.e. $j_D^k \notin \bar{N}^{k+1}$. Otherwise, if after adding i_A^k to $e_{\min}^k$ player j_D^k prefers to switch* *resource $e^k_{\min}$ with another resource $u \notin \sigma_{j^k_D}$, then we show that the S-move can be split into two moves, a D-move and an A-move. Note that $l_u(h_u+1) < l_{e^k_{\min}}(h_{e^k_{\min}}+1)$ implies $u < e^k_{\min}$, and therefore, at the next iteration, j^k_D will be the single player in the set $\{i \in \bar{N}^{k+1} : e^i = e^{k+1}_{\min}\}$. Hence, at the next iteration $(k+1)$, player j^k_D will be selected by the algorithm as player i^{k+1}_A and he will add resource u to his strategy. Thus, splitting the S-move into a D-move and an A-move does not effect the process.*

Proof of Lemma 3.8: Consider player $p = i^k_A$ who adds resource $e^k_{\min}$ to his strategy σ^k_p at the beginning of the k'th iteration. For any $e \leq e^k_{\min}$, we prove below that $e \in \sigma^r_p$ for all $r > k$ such that $h^r = h^k$. Since $e^k_{\min} \notin \sigma^k_p$, we get $\sigma^r \neq \sigma^k$ for all such r, implying there are no cycles in the TNE-algorithm.

Assume on the contrary that player p removes some resource $e \leq e^k_{\min}$ from his strategy set, before or at the r'th iteration. Let $k < s \leq r$ be the first iteration at which such a D-move is applied. Then, this change is caused by an A-move of player $q = i^s_A$ with resource $e^s_{\min} \in \sigma^s_p$. Let $\sigma^{s+} = (\sigma^s_{-q}, \sigma^s_q \cup \{e^s_{\min}\})$. Since by the algorithm every $e \leq e^s_{\min}$ satisfies $e \in \sigma^s_p$, player p cannot improve his payoff by switching resource $e^s_{\min}$ with another resource, but only by removing $e^s_{\min}$ from σ^s_p. Then, $C_p(\sigma^{s+}) > C_p(\sigma^{s-})$, where $\sigma^{s-} = (\sigma^{s+}_{-p}, \sigma^{s+}_p \setminus \{e^s_{\min}\})$. By Observation 3.4(1), this implies

$$l_{e^s_{\min}}(h_{e^s_{\min}} + 1) > w_p - \frac{t}{sf^{|\sigma^{s+}_p|-1}}. \tag{16}$$

Let $k \leq l < s$ be the last iteration where player p adds a resource to his strategy, before dropping resource $e^s_{\min}$ (recall that player p applies an A-move at the k'th iteration). Then, $C_p(\sigma^{l+}) \leq C_p(\sigma^l)$, where $\sigma^{l+} = (\sigma^l_{-p}, \sigma^l_p \cup \{e^l_{\min}\})$. By Observation 3.4(2), this implies

$$l_{e^l_{\min}}(h_{e^l_{\min}} + 1) \leq w_p - \frac{t}{sf^{|\sigma^l_p|}}. \tag{17}$$

Since $|\sigma^{s+}_p| \leq |\sigma^l_p| + 1$, (16) and (17) imply

$$l_{e^s_{\min}}(h_{e^s_{\min}} + 1) > w_p - \frac{t}{sf^{|\sigma^{s+}_p|-1}} \geq w_p - \frac{t}{sf^{|\sigma^l_p|}} \geq l_{e^l_{\min}}(h_{e^l_{\min}} + 1),$$

in contradiction to $l_{e^s_{\min}}(h_{e^s_{\min}} + 1) \leq l_{e^l_{\min}}(h_{e^l_{\min}} + 1)$. □

4 Symmetric TCGFs

In this section we consider symmetric TCGFs. In a symmetric TCGF, the parameters of the game are not a function of the player or the resource identities, i.e. for all $i \in N$ and

$e \in M$ we have $w_i = w$ and $l_e(k) = l(k)$ for all $k \in \{0, 1, \ldots, n\}$. Clearly, the TNE-algorithm is valid for symmetric TCGFs. However, for these, relatively simple, games, we present a significantly simpler algorithm – the STNE-algorithm – which easily finds a pure strategy Nash equilibrium profile.

The algorithm is initialized with an empty strategy for each player. It orders the set $N \times M = \{(i, e) \; : \; i \in N, e \in M\}$ of pairs of players and resources, according to the rule described in the sequel. Then, using this order, it offers the players a resource to be added to their strategy. If the corresponding A-move of resource e to strategy σ_i of player i does not deteriorate his payoff, the algorithm updates his strategy and proceeds to the next pair. The algorithm halts upon the first decline. The STNE-algorithm is presented below. For simplicity of notation, we denote $a(\text{mod } b)$ by $[a]_b$.

STNE-algorithm

Initialization: For all $i \in N$, set $\sigma_i := \varnothing$;
Set $k := 0$;

Main step: 1. Set $k := k + 1$.
If $k > \gcd(m, n)$, then QUIT;

2. Set $q := 1$;

(a) Let $e(q) = e_{[q+k-1]_m}$;

(b) If $C_{[q]_n}(\sigma_{-[q]_n}, \sigma_{[q]_n} \cup \{e(q)\}) \leq C_{[q]_n}(\sigma)$
then set $\sigma_{[q]_n} := \sigma_{[q]_n} \cup \{e(q)\}$;
Otherwise, QUIT;

(c) Set $q := q + 1$. If $q > \text{lcm}(m, n)$
then go to 1. Otherwise, go to (a).

The procedure of ordering the set $N \times M$ is illustrated by the following example. Suppose there are $n = 9$ players and $m = 6$ resources. We define an order in which we offer the players to add a resource to their strategy as follows (see Figure 3). We assign the players to the resources by first assigning player 1 to resource e_1, then player 2 is assigned to resource e_2, and so on until player 6 is assigned to the last resource – e_6. Then we continue with player 7 going to resource e_1, player 8 to e_2, and the last player – player 9 – gets resource e_3. We start a new sequence by assigning player 1 to resource e_4, and so on until at the end of the first

e_1	e_2	e_3	e_4	e_5	e_6
1	2	3	4	5	6
7	8	9	1	2	3
4	5	6	7	8	9
	1	2	3	4	5
6	7	8	9	1	2
3	4	5	6	7	8
9					
		1	2	3	4
5	6	7	8	9	1
2	3	4	5	6	7
8	9				

Figure 3: Example for the implementation of the STNE-algorithm.

iteration, player 9 is assigned to resource e_6. At the next iteration, we start a new sequence by moving the players by one step: namely, player 1 is assigned to resource e_2, player 2 is assigned to resource e_3, and at the end of the iteration, player 9 is assigned to resource e_1. The length of each iteration is bounded by the least common multiplier of m and n, and the number of iterations is bounded by the greatest common divider of m and n.

Theorem 4.1 *The STNE-algorithm finds a pure strategy Nash equilibrium in a given symmetric TCGF in O(nm).*

Proof: First, we show that the STNE-algorithm does not assign a player to a particular resource more than once and thus provides a feasible assignment. Assume this is not true. Then, either there exist q_1, q_2 such that $q_1+k-1 \equiv q_2+k-1 (\text{mod } m)$, where $q_1 \equiv q_2 (\text{mod } n)$, or there are k_1 and k_2 such that $q_1+k_1-1 \equiv q_2+k_2-1 (\text{mod } m)$, where $q_1 \equiv q_2 (\text{mod } n)$. In the first case, $q_1+k-1 \equiv q_2+k-1 (\text{mod } m) \Rightarrow q_1 \equiv q_2 (\text{mod } m)$. That is, $q_1 - q_2$ divides both m and n, and therefore $q_1 - q_2 \geq \text{lcm}(m,n)$. But since $q_1 \leq \text{lcm}(m,n)$ and $q_2 \geq 1$, we get $q_1 - q_2 < \text{lcm}(m,n)$, a contradiction. In the second case, $q_1+k_1-1 \equiv q_2+k_2-1 (\text{mod } m)$ $\Rightarrow q_1 - q_2 \equiv k_2 - k_1 (\text{mod } m)$. That is, $k_2 - k_1$ divides $\text{gcd}(m,n)$, and therefore $k_2 - k_1 \geq \text{gcd}(m,n)$. But since $k_2 \leq \text{gcd}(m,n)$ and $k_1 \geq 1$, we get $k_2 - k_1 < \text{gcd}(m,n)$, a contradiction.

The above also implies that the time complexity of the algorithm is $O(nm)$, since a player is assigned to a resource at most once, without reordering neither players nor resources. One can also notice that the length of each iteration of the algorithm is bounded by $\mathrm{lcm}(m, n)$, and the number of iterations is bounded by $\gcd(m, n)$, leading to the same bound of complexity of the algorithm.

Next we show that the output of the STNE-algorithm is a Nash equilibrium strategy profile. First, we prove that the output of each step of the algorithm is DS-stable. Then we show that the algorithm terminates with an A-stable strategy profile. Thus, by the single profitable move property, the resulting combination of strategies is a Nash equilibrium.

Let σ^r be the output of the r'th step of the STNE-algorithm, and let h^r denote the corresponding congestion vector ($h^r = h(\sigma^r)$). We show below that σ^r is DS-stable. The proof uses the induction principle. For $r = 1$ the proof is immediate. At the first step, the algorithm offers player 1 to add resource e_1 to his strategy. If the algorithm receives decline, then the resulting strategy profile of the first step is $\sigma^1 = (\varnothing, \ldots, \varnothing)$ which is obviously DS-stable. If player 1 adds resource e_1 to his strategy then

$$C_1(\{e_1\}, \varnothing, \ldots, \varnothing) = wf + sl(1) < w = C_1(\varnothing, \ldots, \varnothing).$$

This implies that player 1 does not wish to drop resource e_1. By the symmetry between resources, player 1 does not want to switch resource e_1 with any other resource. That is,

$$C_1(\{e_1\}, \varnothing, \ldots, \varnothing) = wf + sl(1) = C_1(\{e\}, \ldots, \varnothing),$$

for all $e \in M$. Other players have nothing to drop or exchange. Thus, σ^1 is DS-stable.

Now we prove our statement for $r > 1$. Assume that at the r'th step the algorithm offers player i to add resource a to his strategy. If the algorithm receives decline, then $\sigma^r = \sigma^{r-1}$, and σ^r is DS-stable by induction. Otherwise, player i adds resource a to his strategy. Then,

$$C_i(\sigma_{-i}^{r-1}, \sigma_i^{r-1} \cup \{a\}) = C_i(\sigma^r) \leq C_i(\sigma^{r-1}) = C_i(\sigma_{-i}^r, \sigma_i^r \setminus \{a\}). \tag{18}$$

We show below that no player wishes to apply a D-move from σ^r; that is, $C_j(\sigma^r) \leq C_j(\sigma_{-j}^r, \sigma_j^r \setminus \{e\})$, for all $j \in N$, $e \in \sigma_j$. Note that $h_a^r \geq h_e^r$ for all $e \in M$. Then, by (18) and Observation 3.4(1), we get $l(h_e^r) \leq w - \frac{t}{sf^{|\sigma_i^r|-1}}$ for all $e \in \sigma_i^r$. Now, since for all $j \in N$ the STNE-algorithm satisfies $|\sigma^r_j| \leq |\sigma^r_i|$, for any $j \in N$ and $e \in \sigma^r_j$ the above yields $l(h^r_e) \leq w - \frac{t}{sf^{|\sigma^r_j|-1}}$, implying by Observation 3.4(1) the non-profitability of a D-move of j from σ^r. Thus, since no player in N wishes to apply a D-move, σ^r is D-stable.

We note that the STNE-algorithm satisfies $h^r_e \leq h^r_{e'} + 1$ for all $e, e' \in M$. Then, Observation 3.3 implies that $C_j(\sigma^r) \leq C_j(\sigma^r_{-j}, (\sigma^r_j \setminus \{e\}) \cup \{e'\})$ holds for all $j \in N$, $e \in \sigma^r_j$, $e' \notin \sigma^r_j$. That is, no player wishes to apply an S-move from σ^r, implying that σ^r is S-stable.

It remains to show that the last iteration of the STNE-algorithm produces an A-stable strategy profile, σ. Assume that at the last iteration the algorithm offers player i to add resource a to his strategy, and receives decline. Then, by Observation 3.4(2), $l(h_a(\sigma)+1) > w - \frac{t}{sf^{|\sigma_i|}}$. Now, since the STNE-algorithm satisfies $h_e \geq h_a$ for all $e \in M$ and $|\sigma_j| \geq |\sigma_i|$ for all $j \in N$, the above yields $l(h_e(\sigma)+1) > w - \frac{t}{sf^{|\sigma_j|}}$ for all $e \in M$, $j \in N$. Recall that $h_e(\sigma)+1 \geq h_{e'}(\sigma)$ for all $e, e' \in M$. Thus, by Observation 3.4(2), the above implies the non-profitability of an A-move from σ.

Thus, the resulting strategy profile of the STNE-algorithm is A-, D- and S-stable. By Lemma 3.5, this strategy profile is a Nash equilibrium. $\square$

5 Summary and Future Research

In this paper, we introduced and studied congestion settings with unreliable resources, in which resource usage is costly. This study is motivated by a variety of situations in which a fixed payment for utilizing resources is demanded by their owners or, alternatively, there is some central coordinator that imposes taxes in order to achieve better social results. We defined the class of taxed congestion games with failures [TCGFs] which refers to congestion effects, resource failures, and costly submission, in a unified game-theoretic setting. Our model extends on the model presented in [8] by considering submission costs (taxes). We proved that TCGFs possess pure strategy Nash equilibria, despite the non-existence of a potential function. Our proof is constructive and yields an efficient non-trivial procedure for constructing such equilibria in these games. We also introduced a simplified efficient algorithm for the case of symmetric TCGFs.

Future research may evaluate the (in)efficiency of Nash equilibria in TCGFs (e.g. the price of anarchy, the price of stability etc.) and develop methods for improving the social outcome obtained by selfish players. In this context, it may be of interest to formulate meaningful conditions under which resource taxation can reduce the total cost suffered in equilibrium. Other future research directions may include the study of the existence of strong equilibrium and coalition-proof equilibrium in TCGFs. While strong equilibrium does not exist in any TCGF, it may be of interest to find cases when it exists; the study of the existence of coalition-proof equilibrium in TCGFs is a pending complementary project.

References

[1] E. Koutsoupias. Selfish task allocation. *Bulletin of EATCS*, 81:79–88, October 2003.

[2] K. Leyton-Brown and M. Tennenholtz. Local-effect games. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 772–777, 2003.

[3] I. Milchtaich. Congestion games with player-specific payoff functions. *Games and Economic Behavior*, 13:111–124, 1996.

[4] I. Milchtaich. Congestion models of competition. *American Naturalist*, 147(5):760–783, 1996.

[5] D. Monderer. Solution-based congestion games. *Advances in Mathematical Economics*, 8:397–407, 2006.

[6] D. Monderer and L. Shapley. Potential games. *Games and Economic Behavior*, 14:124–143, 1996.

[7] A. Orda, R. Rom, and N. Shimkin. Competitive routing in multi-user communication networks. In *IEEE/ACM Transactions on Networking*, volume 1, pages 510–521, 1993.

[8] M. Penn, M. Polukarov, and M. Tennenholtz. Congestion games with failures. In *Proceedings of the 6th ACM Conference on Electronic Commerce (EC-05)*, pages 259–268, 2005.

[9] M. Penn, M. Polukarov, and M. Tennenholtz. Congestion games with load-dependent failures: identical resources. In *Proceedings of the 8th ACM Conference on Electronic Commerce (EC-07)*, pages 210–217, 2007.

[10] T. Quint and M. Shubik. *A model of migration*. Number 1088 in Cowles Foundation Discussion Papers. Cowles Foundation, Yale University, 1994.

[11] R. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.