Strong Price of Anarchy*

Nir Andelman† Michal Feldman‡ Yishay Mansour§

Abstract

A strong equilibrium (Aumann 1959) is a pure Nash equilibrium which is resilient to deviations by coalitions. We define the strong price of anarchy to be the ratio of the worst case strong equilibrium to the social optimum. In contrast to the traditional price of anarchy, which quantifies the loss incurred due to both selfishness and lack of coordination, the strong price of anarchy isolates the loss originated from selfishness from that obtained due to lack of coordination. We study the strong price of anarchy in two settings, one of job scheduling and the other of network creation. In the job scheduling game we show that for unrelated machines the strong price of anarchy can be bounded as a function of the number of machines and the size of the coalition. For the network creation game we show that the strong price of anarchy is at most 2. In both cases we show that a strong equilibrium always exists, except for a well defined subset of network creation games.

1 Introduction.

Much of the classical work in scheduling and optimization has been centered on finding efficient algorithms, in the sense that they optimize a certain global function (also called *social optimum*). The recent interest in computational game theory is based, in part, on the recognition that the agents involved may be selfish, meaning that they are motivated by optimizing their own utilities rather than reaching the social optimum. Therefore, one cannot assume that the social optimum can be enforced on the selfish agents.

A natural concern is quantifying the efficiency loss incurred due to selfish behavior. A metric that is widely accepted in the computer science literature is the ratio between the worst possible solution reached with selfish agents and the social optimum. The *Price of Anarchy* (PoA) [15, 18] quantifies this loss as the ratio between the cost of the worst Nash equilibrium and the social optimum, and has been extensively studied in the contexts of the contexts of selfish routing [19], job scheduling [15, 7], network formation [9, 1, 6], and more. In a Nash equilibrium no agent can improve its *own* utility by *unilaterally* changing its action. This notion thus accounts for two properties, namely, selfishness and lack of coordination.

We adopt the solution concept of *strong equilibrium*, proposed by Aumann [3]. In a strong equilibrium, no coalition (of any size) can deviate and improve the utility of *every* member of the coalition (while possibly lowering the utility of players outside the coalition)[1]. Clearly, every strong equilibrium is a Nash equilibrium, but the converse does not hold. In cases where a strong equilibrium exists, it seems to be a very robust notion. Considering strong equilibrium allows us to separate the effect of selfishness (which remains in strong equilibria) from that of lack of coordination (which disappears, since a strong equilibrium is resilient to deviations in coalitions).

We define the *strong price of anarchy* (SPoA) to be the ratio of the worst strong equilibrium and the social optimum. In contrast to the traditional PoA, which quantifies the loss incurred due to both selfishness and lack of coordination, the SPoA isolates the loss that is incurred due to selfishness. The SPoA metric is well defined only when a strong equilibrium exists. Unfortunately, most games do not admit any strong equilibrium (even if mixed strategies are allowed[2]). Thus, in order to analyze the SPoA, one must first prove that a strong equilibrium exists in the specific setting at hand.

In cases where the SPoA yields substantially better results than the PoA, it suggests that coordination can significantly improve the efficiency loss. If, on the other hand, the SPoA and PoA yield similar results, the efficiency loss, as quantified by the PoA, is derived from selfishness alone, and coordination will not help. We will see examples of both cases in the games we study.

*Research partially supported by a grant of the Israel Science Foundation, BSF and an IBM faculty award.

†School of Computer Science, Tel Aviv University.

‡School of Computer Science, Hebrew University of Jerusalem.

§School of Computer Science, Tel Aviv University.

[1]The strong equilibrium solution concept does not require that the deviation itself will be resilient to further deviations. A related, yet weaker, solution concept is a coalition-proof Nash equilibrium [5] requiring that the deviation by the coalition is itself resilient to further deviations by subsets of the coalition. This implies that the coalition-proof Nash equilibrium includes any strong equilibrium but rules out many Nash equilibria.

[2]This is in contrast to Nash equilibrium, which exists (perhaps in mixed strategies) in every finite game.

In our definition we also consider the size of the coalition as a parameter and define k-SPoA to be the ratio of the worst Nash equilibrium which is immune to coalitions of size up to k and the social optimum.[3] This is a natural restriction in many settings, where the ability to coordinate may be limited.

A related, yet different, notion of coalitions has been studied recently by [12], where a coalition is assumed to be fixed and acts as a new selfish player, attempting to maximize its utility. Coalitions have been also considered from the point of view of mechanism design, where it is desired to design truthful mechanisms that are resilient to the formation of coalitions, known as group-strategyproof mechanisms [17, 11].

In this work we consider two different sets of games. The first is derived from job scheduling, where each player controls a single job and selects the machine on which the job is run. The cost to the player is the load on the machine it selected while the social cost is the *makespan* (the maximal load on any machine). The second game is a network creation game [9, 1]. In this game the players can be viewed as nodes in a graph. Each player (node) buys links (to other nodes) at the cost of α per link. The set of edges in the resulting graph is the union of the links that the players (nodes) bought. The cost to the player is the cost of the links it bought plus the sum of the distances to all the nodes (players) in the resulting graph. The social cost is the sum of the players' costs (the social welfare).

For the job scheduling game, we consider mostly the model of unrelated machines (namely, the load of a job is a function of the machine it is scheduled on). While it is rather simple to show that for unrelated machines the PoA is unbounded (see [4]), we show that the SPoA is bounded as a function of the number of players and machines. More specifically, we show that: (1) For m machines the worst-case SPoA is at most $2m-1$ and at least m (and for 2 machines the SPoA is 2.) (2) For m machines and n players the worst-case k-SPoA is at most $O(nm^2/k)$ and at least $\Omega(n/k)$. Moreover, we show that a strong equilibrium always exists, and some optimal solution is also a strong equilibrium.

For the network creation game, we show that for most values of α there is some strong equilibrium. Specifically, for $\alpha \in (0,1]$ we show that the clique is a strong equilibrium and for $\alpha \geq 2$ the star is a strong equilibrium. For $\alpha \in (1,2)$ we show that there is no strong equilibrium in general. More specifically, we show that there is no strong equilibrium when the coalition size is at least 3 and the number of players is at least 6. We show that for either a smaller number of players (four or less) or smaller coalitions (size at most 2) there always exists a strong equilibrium.

Previous work has already bounded the PoA of the network creation game [9, 1]. Roughly, for $\alpha = O(\sqrt{n})$ and $\alpha = \Omega(n \log n)$ the PoA is constant. For $\alpha \in [\sqrt{n}, n]$ the PoA is $O(\alpha^{2/3}/n^{1/3})$ and for $\alpha \in [n, n\log n]$ the PoA is $O(n^{2/3}/\alpha^{1/3})$. We show that for any $\alpha \geq 2$ the SPoA is at most 2.

The *Price of Stability (PoS)* [2] is the ratio of the best Nash equilibrium to the social optimum. Similarly, one can define the *Strong Price of Stability (SPoS)* as the ratio of the best strong equilibrium and the optimum. Our existence results show that for both job scheduling and network creation the SPoS is 1, since there exists an optimal solution which is a strong equilibrium.

The vast literature on strong equilibrium has focused both on pure strategies and pure deviations (e.g., [13, 14, 16, 5]). This has been mainly motivated by the fact that the strong equilibrium is already a solution concept that does not exist in many cases and allowing mixed deviations would only further reduce it. The only exception is [20] where correlated deviations are considered. We show that in the job scheduling setting, once we allow mixed deviations by coalitions, in many cases no strong equilibrium exists (in contrast to pure deviations, where always some strong equilibrium exists). More specifically, in the case of mixed strategies and deviations, for $m \geq 5$ identical machines and $n > 3m$ identical jobs, there is no mixed strong equilibrium with respect to mixed deviations.

2 Model.

In this section we provide general notations and definitions, while in Sections 3.1 and 4.1 we provide the notations and definitions for the specific games we study.

A game is denoted by a tuple $G = < N, (S_i), (c_i) >$, where N is the set of players, S_i is the finite action space of player $i \in N$, and c_i is the cost function of player i.

We denote by $n = |N|$ the number of players. The joint action space of the players is $S = \times_{i=1}^{n} S_i$. For a joint action $s \in S$ we denote by s_{-i} the actions of players $j \neq i$, i.e., $s_{-i} = (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n)$. Similarly, for a set of players Γ we denote by $s_{-\Gamma}$ the actions of players $j \notin \Gamma$. The cost function of player i maps a joint action $s \in S$ to a real number, i.e., $c_i : S \to \mathbb{R}$.

Nash Equilibrium (NE): A joint action $s \in S$ is a *pure* Nash Equilibrium if no player $i \in N$ can benefit from unilaterally deviating from his action to another action, i.e., $\forall i \in N\ \forall a \in S_i\ :\ c_i(s_{-i}, a) \geq c_i(s)$.

Resilience to coalitions: A *pure joint action* of a set of players $\Gamma \subset N$ (also called *coalition*) specifies an

[3] Namely, no coalition of at most k players can coordinate a deviation and improve the utility of every player in the coalition.

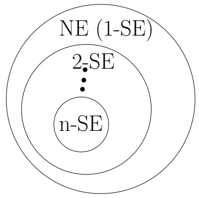


Figure 1: Illustration of the k-SE hierarchy (the set k-SE represents all the NE which are also k-SE).

action for each player in the coalition, i.e., $\gamma \in \times_{i\in\Gamma} S_i$. A joint action $s \in S$ is not resilient to a *pure* deviation of a coalition Γ if there is a pure joint action γ of Γ such that $c_i(s_{-\Gamma}, \gamma) < c_i(s)$ for every $i \in \Gamma$ (i.e., the players in the coalition can deviate in such a way that *each* player reduces its cost). A pure Nash equilibrium $s \in S$ is *resilient to pure deviations of coalitions of size* k, if there is no coalition Γ of size at most k, such that s is not resilient to a pure deviation by Γ.

Definition 2.1. *A* k-strong equilibrium (k-SE) *is a pure Nash equilibrium that is resilient to pure deviations of coalitions of size at most k.*

Clearly, a k-SE is a refinement of NE. Let $\Phi(G, k)$ be the set of k-strong equilibria of the game G. By definition, for any k, $\Phi(G, k) \subseteq \Phi(G, k-1)$ (see Figure 1). Note that $\Phi(G, 1)$ coincides with the set of NE, and $\Phi(G, n)$ coincides with the classical notion of a *strong equilibrium* introduced by Aumann in [3].

Note that while in Nash equilibria we can restrict attention to pure deviations, this is not true for k-strong equilibrium, when $k \geq 2$. The conceptual reason is that we need to guarantee that *each* player in the coalition would benefit from the deviation. In Section 3.4 we show an example in which a coalition can benefit from a mixed deviation, yet in any pure deviation some player in the coalition does not benefit. (We defer the definition of a mixed deviation to the above section.)

In order to study the strong price of anarchy we need to define the *social cost* of a game G. Abstractly, there is a function f_G such that the social cost of $s \in S$ is $f_G(s)$. The optimal social cost is $OPT(G) = \min_{s\in S} f_G(s)$. In the cases discussed in this paper the social cost is a simple function of the costs of the players. More specifically, we discuss the linear case, i.e., $f_G(s) = \sum_{i=1}^{n} c_i(s)$, and the maximum, i.e., $f_G(s) = \max_{i=1}^{n} c_i(s)$. Next we define the strong price of anarchy (SPoA) and strong price of stability (SPoS).

Definition 2.2. *Let $\Phi(G, k)$ be the set of k-strong equilibria of the game G. If $\Phi(G, k) \neq \emptyset$ then:*

1. *the* k-strong price-of-anarchy *(k-SPOA) is the ratio between the* maximal *cost of a k-strong equilibrium and the social optimum, i.e., $(\max_{s\in\Phi(G,k)} f_G(s))/OPT(G)$.*
2. *the* k-strong price-of-stability *(k-SPoS) is the ratio between the* minimal *cost of a k-strong equilibrium and the social optimum, i.e., $(\min_{s\in\Phi(G,k)} f_G(s))/OPT(G)$.*

We denote by $SPoA$ the n-SPoA, and by SPoS the n-SPoS, allowing any size of a coalition. (Note that both SPoA and SPoS are defined only if some strong equilibrium exists.)

Due to space limitations, some of the proofs are only sketched in the appendix, and all the proofs appear in the full version, which can be found in the authors' web sites.

3 Job Scheduling.

In our job scheduling scenario there are m machines and n players (where each player controls a single job). In the job scheduling terminology, we will focus on unrelated machines, but also refer to identical machines.

3.1 Job Scheduling Model. A job scheduling setting is characterized by the tuple $< M, N, (w_i(J)) >$, where $M = \{M_1, \ldots, M_m\}$ is the set of machines, $N = \{1, \ldots, n\}$ is the set of players (jobs) and $w_i(J)$ is the weight of player $J \in N$ on machine $M_i \in M$. A job scheduling setting has identical machines if for every $M_i, M_{i'} \in M$ and $J \in N$, we have $w_i(J) = w_{i'}(J)$. In identical machine settings we will use $w(J)$ to denote the weight of J (on any machine).

A *job scheduling game* has N as the set of players. The action space S_J of player $J \in N$ are all the individual machines, i.e., $S_J = M$. The joint action space is $S = \times_{J=1}^{n} S_J$. In a joint action $s \in S$ player J selects machine s_J as its action. We denote by B_i^s the set of players on machine M_i in the joint action $s \in S$, i.e., $B_i^s = \{J : s_J = M_i\}$. The load of a machine M_i, in the joint action $s \in S$, is the sum of the weights of the players that chose machine M_i, that is $L_i(s) = \sum_{J\in B_i^s} w_i(J)$. For a player $J \in N$, let $c_J(s)$ be the load that player J observes in the joint action s, i.e., $c_J(s) = L_i(s)$, where $s_J = M_i$. A *job scheduling game* is characterized by a tuple $< N, S, (c_J) >$.

In job scheduling games the objective function (i.e., the social cost) is the *makespan*, which is the load on the most loaded machines (or equivalently, the highest load some player observes). Formally, makespan$(s) = \max_J c_J(s)$. A social optimum minimizes the makespan, i.e., $OPT = \min_s$ makespan(s). Thus, the strong price of anarchy (SPoA) in job scheduling games is the ratio between the makespan of the worst SE and the minimal makespan.

Notation: We define $w_{min}(J) = \min_i w_i(J)$, and denote by $\min(J)$ the index of a machine on which player J has weight $w_{min}(J)$, i.e., $\min(J) = \arg\min_i w_i(J)$ (if there is more than one such machine then select an arbitrary one). In addition, we denote by $OPT(J)$ the action of job J under a social optimum OPT.

3.2 Equilibrium Existence. In this section we prove that in the job scheduling game, for any coalitions of size k, there is a k-SE, i.e., there exists a NE that is resilient to coalitions of size k (for any $k \leq n$). Our proof technique is similar to [10, 8], that proved that any sequence of improvement steps, in a job scheduling game, converges to a NE. We first define a complete order on the joint actions.

Definition 3.1. *A vector $(l_1, l_2, \dots l_m)$ is smaller than $(\hat{l}_1, \hat{l}_2, \dots \hat{l}_m)$* lexicographically *if for some i, $l_i < \hat{l}_i$* and *$l_k = \hat{l}_k$ for all $k < i$. A joint action s is* smaller than *s'* lexicographically *if the vector of machine loads $L(s)$, sorted in non increasing order, is smaller lexicographically than $L(s')$, sorted in non increasing order.*

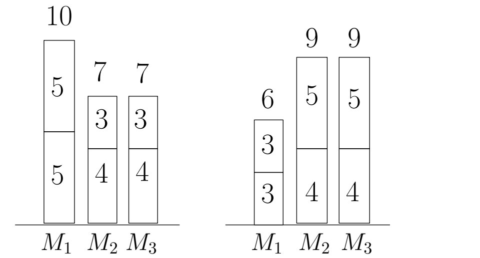
We prove that the lexicographically minimal assignment is a k-SE.

Theorem 3.1. *In any job scheduling game, the lexicographically minimal joint action s is a* k-SE equilibrium*, for any k.*$\square$

An immediate corollary from the fact that a lexicographically minimal joint strategy is a k-SE, is that the k-Strong Price of Stability (k-SPoS) for job scheduling games is 1.

It is shown in [8] that any job scheduling game is a potential game. However, while Theorem 3.1 holds for any job scheduling game, it does not hold in general for any potential game. For example, the prisoner's dilemma game is a potential game, but the only NE in this game (in which both players defect) is not resilient to a coalition of both players cooperating. Thus, the prisoner's dilemma game has no SE.

The requirement that every member in a coalition strictly benefits from the deviation is a crucial assumption for the correctness of Theorem 3.1. If we relax the condition and require only that some member improves its cost and no other member of the coalition would lose from the deviation, there are job scheduling games that do not have any SE. [4]



(a) NE assignment (b) a profitable deviation

Figure 2: An example of an assignment (a) that is a Nash equilibrium but not a strong equilibrium, since the jobs of size $\{5, 5, 3, 3\}$ all profit from the deviation demonstrated in (b).

While the proof technique for the existence of k-SE is similar to [10, 8], it is important to note that there is no equivalence between deviation in coalitions and unilateral deviations. In particular, already with 3 identical machines[5], there exist NE assignments that are not resilient to coalitional deviations, as illustrated in Figure 2.

3.3 Strong Price of Anarchy. In this section we study the SPoA in scenarios with identical and unrelated machines. For identical machines, it is known that PoA ≤ 2 [15], while for unrelated machines, the PoA may be unbounded [4]. Consider the following motivating example for unrelated machines.

Example 3.1. *Consider $m \geq 2$ machines and $n = m$ jobs, where $w_i(J_i) = \epsilon$ for all $1 \leq i \leq m$, and $w_i(J_j) = 1$ for all $i \neq j$. The joint action $(1, 2, \dots, m)$ has a minimal makespan of ϵ (and is also a NE). However, the joint action $(m, 1, 2, \dots, m-1)$ is also a NE and has a makespan of 1. Therefore, the PoA is at least $1/\epsilon$, which can be arbitrarily large. However the only joint action that is resilient to a coalition of all the players is $(1, 2, \dots, m)$, and therefore in this example the SPoA is 1, which is significantly smaller than the PoA.*

Example 3.1 motivates using the SPoA solution concept for unrelated machines. We now prove our main results for the job scheduling games, showing that the strong price of anarchy is bounded in the unrelated ma-

[4]For example, consider the following setting: there are two identical machines, and three identical unit jobs. Clearly, in a NE, a pair of jobs is on one machine and the third job is on the other. However, under the relaxed improvement requirement, no equilibrium is 2-SE: The pair of jobs on the same machine can form a coalition where one job migrates to the other machine, while the other job does not change machines. After the deviation, the migrating job remains with a load of 2, while the load observed by the idle job in the coalition decreases from 2 to 1.

[5]In a job scheduling game with 2 identical machines, one can verify that an assignment is NE if and only if it is SE.

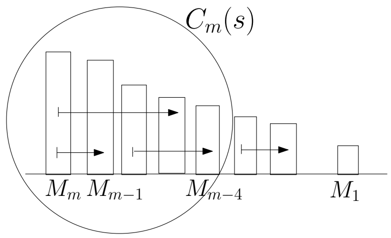


Figure 3: Illustration of C_m.

chine setting. We start with the following straightforward relationship between OPT and the weights.

CLAIM 3.1. *For any job scheduling game with unrelated machines, the following inequalities hold:*

$$OPT \geq \max_J w_{\min}(J) \tag{3.1}$$

$$OPT \geq \frac{1}{m} \sum_J w_{\min}(J) \tag{3.2}$$

where $OPT = \min_{s \in S} \max_i L_i(s)$.

We first bound the SPoA for games with two machines.

THEOREM 3.2. *For any job scheduling game with* 2 *unrelated machines and n jobs, SPoA* ≤ 2.

We next introduce some notations that will be useful. For simplicity, for the rest of this section we will assume WLOG that given a joint action s, the machine indices are sorted in a non-decreasing order of the loads under s, i.e., $L_1(s) \leq \cdots \leq L_m(s)$.

DEFINITION 3.2. *We denote by* $M_i \mapsto_s M_j$, *if there is a job* J *such that* $M_j = \min(J)$, $s_J = M_i$ *and* $i \geq j$. *Two machines* M_i *and* M_j, $i \geq j$, *are* connected under the joint action s *if* $\exists i', j'$, *such that* $i' \geq i$, $j \geq j'$, *and* $M_{i'} \mapsto_s M_{j'}$. *Let* $C_m(s) = \{M_m, \ldots, M_\ell\}$ *denote the maximal suffix of machines, such that* M_{i+1} *is connected to* M_i *under joint action* s. *(See figure 3.)*

By the definition of $C_m(s)$ and the relation $M_i \mapsto_s M_j$ we have,

CLAIM 3.2. *For every job* J *such that* $s_J \in C_m(s)$ *we have* $\min(J) \in C_m(s)$.

The following lemma bounds the difference between loads of machines in $C_m(s)$, under a NE s.

LEMMA 3.1. *Let* s *be a NE. If* $M_i \mapsto_s M_j$ *then* $L_i(s) \leq L_j(s) + OPT$. *In addition, for any* $i, j \in C_m(s)$ *we have* $L_i(s) \leq L_j(s) + (m-1)OPT$.

Proof. Since s is a NE, for each $J \in B_i^s$ we have $L_i(s) \leq L_j(s) + w_j(J)$. From the definition of $M_i \mapsto_s M_j$, there exists $J \in B_i^s$ for which $M_j = \min(J)$. From Inequality (3.1), $w_j(J) \leq OPT$, and we get: $L_i(s) \leq L_j(s) + OPT$.

By consecutive applications of this argument, the load of M_m and M_ℓ, the least loaded machine in C_m, cannot differ by more than $(m-1)OPT$. Therefore, for any two machines M_i and M_j in C_m, $L_i(s) \leq L_j(s) + (m-1)OPT$. $\square$

THEOREM 3.3. *For any job scheduling game with m unrelated machines and n jobs, SPoA* $\leq 2m - 1$.

Proof. Let s be an arbitrary joint action that is a SE. Recall that we assume WLOG that the machines are sorted in a non-decreasing order of the loads.

If for some $M_i \in C_m(s)$ we have $L_i(s) \leq m \cdot OPT$ then by Lemma 3.1 $L_m(s) \leq (2m-1) \cdot OPT$, and we are done. Otherwise, $\forall i \in C_m(s)$, $L_i(s) > m \cdot OPT$. We will show that such a joint action s is not resilient to a deviation of a coalition. Consider the joint action s', where for $J \in C_m(s)$ we have $s'_J = \min(J)$, and for $J \notin C_m(s)$ we have $s'_J = s_J$. This implies that the coalition Γ includes all the jobs scheduled in s on machines in $C_m(s)$, i.e., $\Gamma = \cup_{M_i \in C_m(s)} B_i^s$.

Recall that by Claim 3.2 we have $\min(J) \in C_m(s)$. By Inequality (3.2), $L_i(s') \leq m \cdot OPT < L_i(s)$, for any $M_i \in C_m(s)$. Therefore, each job $J \in C_m(s)$ is strictly better off under s'. $\square$

The following theorem shows that the SPoA might be linear in the number of machines m.

THEOREM 3.4. *There exists a job scheduling game with m unrelated machines for which SPoA* $\geq m$.$\square$

Next, we derive bounds for coalitions whose size is smaller than n. We first present a lower bound for two machines.

THEOREM 3.5. *There exists a job scheduling game with* 2 *unrelated machines and n jobs, s.t. k-SPoA* $\geq \frac{n}{2k}$.

Proof. Consider the following job scheduling game. Let $w_1(J_i) = 1$ and $w_2(J_i) = 1/(n-1)$, for $2 \leq i \leq n$, and let $w_1(J_1) = 2k$ and $w_2(J_1) = n - 1 + k + \epsilon$. In this game $OPT(J_1) = M_1$ and $OPT(J_2) = \cdots OPT(J_n) = M_2$, which yields a cost of $2k$. The joint action $s_1 = M_2$ and $s_2 = \cdots = s_n = M_1$ is a k-SE. (To see that it is a k-SE note that if J_1 migrates to M_1 the new load is $n + 2k$. This implies that at least $k+1$ jobs have to migrate from M_1 in order that it will be beneficial for J_1 to migrate to M_1). Therefore, k-SPoA $\geq \frac{n-1+k+\epsilon}{2k} \geq \frac{n}{2k}$. $\square$

Example 3.1 presents a NE for which the PoA is unbounded. Since the same example is resilient to any coalition of size at most $m-1$, it implies that the $(m-1)$-SPoA is unbounded. The following theorem bounds the k-SPoA for coalitions of size $k \geq m$.

THEOREM 3.6. *For any job scheduling game with m unrelated machines and n jobs, for any $k \geq m$, k-SPoA $\leq \frac{2nm}{z} + 4m$, where $z = \lfloor k/m \rfloor$.*□

For identical machines, we show that the SPoA does not improve on the PoA.

THEOREM 3.7. *There exists a job scheduling game with m identical machines and n jobs, s.t. SPoA $\geq \frac{2}{1+\frac{1}{m}}$.*□

3.4 Mixed Deviations and Mixed Equilibrium.

A natural extension of the SE solution concept would be to consider mixed strategies and deviations. A mixed strategy is a distribution over the action space, and similarly, a mixed coalition deviation assigns a new mixed strategy to every player in the coalition.

If players are only allowed to deviate unilaterally (as in NE), it is known that allowing mixed and pure deviations is equivalent. In contrast to NE, a pure SE might not be preserved when mixed deviations are allowed.[6] We will show that when mixed deviations are allowed, many job scheduling games do not have a SE.

We will use the notation $\pi_J(i)$ to denote the probability that player J chooses machine M_i and let the joint strategy be $\pi = (\pi_1, \ldots, \pi_n)$. The following example shows a pure SE, in a job scheduling game, which is not preserved when mixed deviations are allowed:

EXAMPLE 3.2. *Consider 2 identical machines and 3 unit jobs, J_1, J_2 and J_3. In any NE with pure strategies, two jobs are assigned to one machine, while the third is assigned to the other machine. Clearly, this is also a SE. WLOG, we assume J_1 and J_2 are assigned to M_1, and J_3 to M_2 in s. Consider a coalition Γ consisting of J_1 and J_2, where the mixed deviations are $\pi_1 = \pi_2 = (\frac{3}{4}, \frac{1}{4})$. The original load on M_1 in s is 2. After the deviation, J_1 and J_2 observe an expected load of $1\frac{7}{8}$. Since both players improve their costs, there is no pure NE that is a 2-SE.*

Although Example 3.2 shows that there is no pure SE when mixed deviations are allowed, in the above example there is a mixed SE.[7] However, in many cases allowing mixed deviations by a coalition eliminates *all* NE. The following theorem proves that this occurs even for identical machines and unit size jobs.

THEOREM 3.8. *For $m \geq 5$ identical machines and $n > 3m$ unit jobs, there is no 4-SE when mixed deviations are allowed.*

Theorem 3.8 required that the coalitions would be of size 4, in order to demonstrate deviations with unit size jobs. The following theorem shows that with weighted jobs, there are settings where even coalitions of size as small as 2 eliminate all NE.

THEOREM 3.9. *There exists a job scheduling game with 2 identical machines and 3 jobs, where no joint mixed strategy is a 2-SE, when mixed deviations are allowed.*□

4 Network Creation.

In this section we study a network creation game which was introduced by [9]. The game models the tradeoff of the agents (nodes) between buying links (edges) and reducing the distances to other nodes. In this section we discuss both the existence of a SE and the SPoA.

4.1 Network Creation Model.

In the network creation game, there are n players, each of which is associated with a separate network vertex. The players buy edges to other nodes and the resulting network is an undirected graph. The cost of each player consists of two components. First, a player pays a cost of $\alpha > 0$ per edge it buys. Second, a player incurs a distance cost equal to the sum of the distances to the other nodes.

Formally, we represent the set of players by a vertex set $V = \{1, \ldots, n\}$. For a player $v \in V$, an action $s_v \in S_v$ is a subset of the edges that include v, i.e., $s_v \subset \{(v,u) | u \in V \setminus \{v\}\}$. The action set of player v is S_v, which is the union of all the possible actions s_v.

Given a joint action $s = (s_1, \ldots, s_n)$, let the resulting graph $G(s) = (V, E)$ consist of the edge set $E = \bigcup_{v \in V} s_v$. Let $\delta_s(v, w)$ be the length of the shortest path between v and w in $G(s)$.

The cost for a player v under joint action s is $c_v(s)$, and is composed from two parts. The *buying cost* is $B_s(v) = \alpha |s_v|$, which charges α for each edge v buys. The *distance cost* is $Dist_s(v) = \sum_{w \in V} \delta_s(v, w)$. The cost for player v is $c_v(s) = B_s(v) + Dist_s(v)$. When clear from the context we will omit the subscript s and use $\delta(v, w)$, $B(v)$, and $Dist(v)$ rather than $\delta_s(v, w)$, $B_s(v)$, and $Dist_s(v)$, respectively.

For a joint action $s \in S$, let the social cost be the total cost of all players, i.e., $cost(s) = \sum_{v \in V} c_v(s)$, and the optimal social cost is $OPT = \min_{s \in S} cost(s)$.

[6] Rozenfeld and Tennedholtz [20] consider an even stronger solution concept of correlated equilibria, and have shown that in a congestion game, it is possible that there is no strong correlated equilibrium in mixed strategies.

[7] The SE has $\pi_1 = (1, 0)$, $\pi_2 = (0, 1)$ and $\pi_3 = (1/2, 1/2)$.

Remark: In our analysis it will sometimes be convenient to assume that the edges have a direction. A directed edge (v, w) indicates that player v buys an edge to w.

4.2 Equilibrium Existence. It was shown in [9] that for $\alpha < 1$ the clique is the social optimum and also the unique NE. For $1 < \alpha < 2$, the clique is the social optimum, but it is no longer a NE, and the star is the worst NE. Finally, for $\alpha \geq 2$, the star is the social optimum, and also a NE, but not a unique one. In this section we analyze the existence of SE for the different values of α. Our main positive result is that for any $\alpha \geq 2$ there is a SE.

THEOREM 4.1. *Let s^* be a joint action where $s^*_r = \emptyset$ and $s^*_v = \{(v, r)\}$, for $v \neq r$ (i.e., $G(s^*)$ is a star in which all the nodes buy edges to the root r). For $\alpha \geq 2$, the joint action s^* is a SE.*

Proof. For contradiction, assume there exists a coalition Γ and a deviation s', in which all nodes in Γ strictly gain from a deviation to s'. Clearly, $r \notin \Gamma$, since in s^* the root r has the lowest possible cost (it does not buy any edges and enjoys the minimum possible distance cost, i.e., distance of 1 to all nodes). For any node $v \in \Gamma$, let x_v denote the number of its *new outgoing* edges, and y_v denote the number of its *new incoming* edges. Obviously, all the new edges originate from nodes in the coalition. Thus, it must hold that $\sum_{v\in\Gamma} x_v \geq \sum_{v\in\Gamma} y_v$. We separate the analysis to two cases:

Case (a): There exists a node v for which $x_v > y_v$. If v does not remove its original edge to r, the change in v's cost is $\alpha x_v - (x_v + y_v) \geq \alpha x_v - 2x_v + 1$ which is positive for $\alpha \geq 2$ (which implies that the cost of v increased). If v removes its edge to r, the change in v's cost is $\alpha x_v - (x_v + y_v) - \alpha + 1 \geq \alpha x_v - 2x_v + 2 - \alpha = (x_v - 1)(\alpha - 2) \geq 0$, since $x \geq 1$ and $\alpha \geq 2$.

Case (b): For every $v \in \Gamma$, $x_v = y_v$. If v does not remove its original edge to r, $B(v)$ increases by αx_v, and $Dist(v)$ decreases by $x_v + y_v$. Therefore, v's cost change is $\alpha x_v - (x_v + y_v) = (\alpha - 2)x_v \geq 0$, since $\alpha \geq 2$. Thus, if $x_v = y_v$, v may improve its cost only if it removes the edge to r. However, if all the nodes in Γ remove their edges to r, the only way for v to remain connected to r (to prevent a distance cost of ∞) is to buy an edge to a node $u \notin \Gamma$. In such a case, $\sum_{v\in\Gamma} x_v > \sum_{v\in\Gamma} y_v$, hence, there exists a node $v \in \Gamma$ for which $x_v > y_v$.
In each case, some $v \in \Gamma$ does not strictly gain from joining the coalition, and therefore s^* is a SE. $\square$

Theorem 4.1 shows that for $\alpha \geq 2$, there exists a star that is a SE. Similarly, we can show that a star in which the root buys edges to all the nodes is also a SE. We conjecture that for $\alpha \geq 2$, *any* star is a SE, regardless of how the edges are bought (we can prove this conjecture only for $\alpha \geq n - 2$).

For $\alpha < 1$, we establish the following:

THEOREM 4.2. *For $\alpha < 1$, s is a SE iff $G(s)$ is a clique. For $\alpha = 1$, if $G(s)$ is a clique, then s is a SE.*

Proof. For $\alpha < 1$ every NE is a clique [9], which implies that if s is a SE then $G(s)$ is a clique. For the other direction (which applies to $\alpha \leq 1$), consider a joint action s such that $G = G(s)$ is a clique. Suppose that there exists a coalition Γ that deviates to s', such that the obtained graph is $G' = G(s')$, which is possibly not a clique. Let x denote the number of edges that are "missing" from the clique, i.e., $x = |E_G| - |E_{G'}|$. (If G' is a clique then $x = 0$.) For each missing edge, there exists a node $v \in \Gamma$ whose buying cost, $B(v)$, decreased by $\alpha \leq 1$. Thus $\sum_{v\in\Gamma} B(v)$ decreased by exactly $\alpha x \leq x$. However, for each missing edge, there exists at least one node in Γ whose distance cost increased by 1. Thus, $\sum_{v\in\Gamma} Dist(v)$ increased by at least x. Therefore, the sum of the costs for nodes in the coalition has not decreased. Therefore, there exists a node $u \in \Gamma$ such that $B(u) + Dist(u)$ has not decreased. In contradiction to the assumption that every $v \in \Gamma$ gains from the deviation to s'. $\square$

An immediate corollary from Theorems 4.1 and 4.2 is that for any $\alpha \notin (1, 2)$ we have $SPoS = 1$.

We next show that for $\alpha \in (1, 2)$ there is no SE (even if we limit the coalition size to 3).

THEOREM 4.3. *For any $\alpha \in (1, 2)$, and any $n \geq 7$, there does not exist any 3-SE.*

The proof of Theorem 4.3 is quite involved. In the following, we will attempt to give a very high level view of the proof. Consider a graph $G(s)$ that has an independent set of size at least 3. We can build a coalition composed of three nodes from the independent set, each buying one edge (and thus forming a triangle). Each node paid $\alpha < 2$ and its distance to the other two nodes is reduced by at least 2. Therefore, all the three nodes gain from this deviation. So our first observation is that in any 3-SE there cannot exist an independent set of size 3 (Lemma B.1). Next we show that there cannot exist any triangle in $G(s)$ (Lemma B.3). Based on those two lemmas, we show that the degree of each node must be at least $n - 3$ (Lemma B.4). Finally, we show that in such a graph, the removal of any edge is beneficial to its buyer.

To complete the analysis for $\alpha \in (1, 2)$, it is easy to see that for $n = 2$ any single edge is a SE, and for

$n = 3$ any tree is a SE. In addition, one can verify that for $n = 4$, any ring in which each node buys a single edge is a SE. For $n = 5, 6$, we show that there does not exist any SE. Interestingly enough, while coalitions of size 3 or more excludes any SE, we show that 2-SE do exist for any number of players.

4.3 Strong Price of Anarchy. In this section we bound the SPoA for $\alpha \geq 2$. The analysis for smaller values of α is trivial [8].

Similarly to [1], we first show that the PoA is dominated by the distance cost.

LEMMA 4.1. *Let s be a NE. For any node v we have $cost(s) \leq (n-1)(2\alpha + n - 1 + Dist(v))$.*□

Our main result is the following.

THEOREM 4.4. *For any $\alpha \geq 2$ and any n, we have SPoA≤ 2.*

The proof of Theorem 4.4 follows directly from the next two lemmas.

LEMMA 4.2. *Let s be a NE. Assume that for every node v, such that $s_v \neq \emptyset$, we have that $Dist(v) > 3n - 5$. Then s is not a SE.*

Proof. Let Γ be the set of all nodes v that bought some edge in s, i.e., $\Gamma = \{v | s_v \neq \emptyset\}$. We will show that Γ can deviate, such that all its members would benefit from a deviation. In the deviation we build a tree T in which each node in Γ buys at most the same number of edges as in s and it strictly reduces the distances to other nodes, i.e., every $v \in \Gamma$ lowers its cost in the deviation.

Assume that there is some node $r \notin \Gamma$. Let T be the following tree. The root of the tree is r. The nodes in the first level are the nodes in Γ. The nodes in the second level are the remaining $n - |\Gamma| - 1$ nodes. Each node in Γ buys an edge to the root r and at most $|s_v| - 1$ edges to nodes in the second level (the leaves). Clearly the number of edges that each node in Γ bought can only decrease. To see that we have enough edges to connect all the $n - |\Gamma| - 1$ leaves, note that in s at least $n - 1$ edges are bought (otherwise some node is disconnected, and all the nodes have infinite cost). We need only $n - 1$ edges to connect all the nodes in T, so we have a sufficient number of edges.

Fix a node $v \in \Gamma$. The distances $Dist(v)$ in T is at most $1 + 2(|\Gamma| - 1) + 3(n - |\Gamma| - 1) \leq 3n - 5$, since $|\Gamma| \geq 1$. Hence, node v improved on its distance cost in s and did not increase its buying cost. Therefore, in this case, s is not a SE.

In the case in which there is no $r \notin \Gamma$ we can select any node to be the root and the remaining nodes will buy an edge to it. Since all the nodes bought at least one edge, the cost of buying edges can only decrease per node. The distances of a node v is now at most $2(n-2) + 1 \leq 3n - 5$ for $n \geq 2$, hence v improved on its distance cost in s.□

LEMMA 4.3. *Let s be a NE. Assume that for some node v, such that $s_v \neq \emptyset$, we have that $Dist(v) \leq 3n - 5$. Then $\frac{cost(s)}{cost(OPT)} \leq 2$.*

Proof. By Lemma 4.1 we have that

$$\begin{aligned} cost(s) &\leq (n-1)(2\alpha + n - 1 + Dist(v)) \\ &\leq (n-1)(2\alpha + n - 1 + 3n - 5) \\ &= 2(n-1)(\alpha + 2n - 3) \end{aligned}$$

For OPT we have: $cost(OPT) = \alpha(n-1) + (n-1)(2(n-2)+1) + (n-1) = (n-1)(\alpha + 2n - 2)$ and the ratio is at most 2.□

Acknowledgements. We thank Noam Nisan for helpful comments on an earlier draft of the paper.

References

[1] S. Albers, S. Elits, E. Even-Dar, Y. Mansour, and L. Roditty. On Nash Equilibria for a Network Creation Game. In *seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2006.

[2] E. Anshelevich, A. Dasgupta, J. M. Kleinberg, É. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. In *FOCS*, pages 295–304, 2004.

[3] R. Aumann. Acceptable Points in General Cooperative n-Person Games. In *Contributions to the Theory of Games*, volume 4, 1959.

[4] B. Awerbuch, Y. Azar, Y. Richter, and D. Tsur. Tradeoffs in Worst-Case Equilibria. In *1st International Workshop on Approximation and Online Algorithms*, 2003.

[5] D. B. Bernheim, B. Peleg, and M. D. Whinston. Coalition-proof nash equilibria: I concepts. *Journal of Economic Theory*, 42:1–12, 1987.

[6] J. Corbo and D. Parkes. The price of selfish behavior in bilateral network formation. In *ACM Symposium on Principles of Distributed Computing*, pages 99–107, 2005.

[7] A. Czumaj and B. Vöcking. Tight bounds for worst-case equilibria. In *SODA*, pages 413–420, 2002.

[8] Recall that for $\alpha < 1$ the clique is the only SE. For $\alpha = 1$, it is easy to see that PoA < 2, since in any NE the distance between any two nodes cannot exceed 2. For $\alpha \in (1, 2)$ we do not have any SE for $n \geq 5$.

[8] E. Even-Dar, A. Kesselman, and Y. Mansour. Convergence time to nash equilibria. In *ICALP*, pages 502–513, 2003.

[9] A. Fabrikant, A. Luthra, E. Maneva, C. Papadimitriou, and S. Shenker. On a network creation game. In *ACM Symposium on Principles of Distriubted Computing (PODC)*, 2003.

[10] D. Fotakis, S. Kontogiannis, E. Koutsoupias, M. Mavronicolas, and P. Spirakis. The structure and complexity of Nash equilibria for a selfish routing game. In *ICALP*, pages 123–134. Springer-Verlag, 2002.

[11] A. V. Goldberg and J. D. Hartline. Collusion-resistant mechanisms for single-parameter agents. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 620–629, 2005.

[12] A. Hayrapetyan, E. Tardos, and T. Wexler. The Effect of Collusion in Congestion Games. In *38th ACM Symposium on Theory of Computing*, 2006.

[13] R. Holzman and N. Law-Yone. Strong equilibrium in congestion games. *Games and Economic Behavior*, 21:85–101, 1997.

[14] R. Holzman and N. L.-Y. (Lev-tov). Network structure and strong equilibrium in route selection games. *Mathematical Social Sciences*, 46:193–205, 2003.

[15] E. Koutsoupias and C. H. Papadimitriou. Worst-case equilibria. In *STACS*, pages 404–413, 1999.

[16] I. Milchtaich. Crowding games are sequentially solvable. *International Journal of Game Theory*, 27:501509, 1998.

[17] H. Moulin and S. Shenker. Strategyproof sharing of submodular costs: Budget balance versus efficiency. *Economic Theory*, 18:511–533, 2001.

[18] C. H. Papadimitriou. Algorithms, games, and the internet. In *proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 749–753, 2001.

[19] T. Roughgarden and E. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236 – 259, 2002.

[20] O. Rozenfeld and M. Tennenholtz. Strong and correlated strong equilibria in monotone congestion games. In *WINE*, 2006.

A Job Scheduling.

Theorem 3.2 *For any job scheduling game with 2 unrelated machines and n jobs, SPoA ≤ 2.*

Proof. Let s be a SE and, WLOG, $L_2(s) \geq L_1(s)$. In the case that for every $J \in B_2^s$ we have $w_2(J) \leq w_1(J)$, by Inequality (3.2), $L_2(s) \leq 2OPT$, and we are done. Otherwise, there exists some $J \in B_2^s$ such that $w_2(J) > w_1(J)$. Since s is a SE, it is in particular a NE, which means that no job on M_2 can gain by unilaterally migrating to M_1. Therefore, $L_2(s) \leq L_1(s) + w_1(J)$. By Inequality (3.1), we get:

$$L_2(s) \leq L_1(s) + OPT \tag{1.3}$$

The following are the possible cases relating OPT, $L_1(s)$ and $L_2(s)$:

1. if $L_1(s) \leq L_2(s) < OPT$, this is impossible (a contradiction to the minimality of OPT).
2. if $OPT < L_1(s) \leq L_2(s)$, then s is not resilient to a coalition of size n (since by deviating to OPT all the players strictly gain).
3. If $L_1(s) \leq OPT \leq L_2(s)$, then from Inequality (1.3), we get: $L_2(s) \leq L_1(s) + OPT \leq 2OPT$.

Taking the maximum over all cases, we get: SPoA ≤ 2. $\square$

Mixed Deviations. In the remainder of this appendix we discuss the case of mixed deviations. We start with the following lemma which greatly limits the structure of a mixed SE.

LEMMA A.1. *Given a k-SE with mixed strategies π, for some $k \geq 2$, let J_1 and J_2 be two jobs with strictly mixed strategies. The supports of π_1 and π_2 must be disjoint.* $\square$

Theorem 3.8 *For $m \geq 5$ identical machines and $n > 3m$ unit jobs, there is no 4-SE, if mixed deviations are allowed.*

Proof. We first consider equilibria with pure strategies. Since all jobs are unit sized, the only equilibrium with pure strategies is when the load on each machines is either $\lfloor \frac{n}{m} \rfloor$ or $\lceil \frac{n}{m} \rceil$.

Let $k = \lceil \frac{n}{m} \rceil$. Since $n > 3m$, there exists a machine with at least 4 jobs assigned to it. WLOG, assume M_1 is one of these machines, and J_1, J_2, J_3, J_4 are four of the jobs that chose it.

Consider the following mixed deviation of these jobs:

$$\pi_1 = \left(\frac{1}{2}, \frac{1}{2}, 0, 0, 0, 0, \ldots, 0\right)$$
$$\pi_2 = \left(\frac{1}{2}, 0, \frac{1}{2}, 0, 0, 0, \ldots, 0\right)$$
$$\pi_3 = \left(\frac{1}{2}, 0, 0, \frac{1}{2}, 0, 0, \ldots, 0\right)$$
$$\pi_4 = \left(\frac{1}{2}, 0, 0, 0, \frac{1}{2}, 0, \ldots, 0\right)$$

The strategies of the remaining jobs are unchanged. In the original joint strategy, the load observed by each of these jobs is k. The expected load observed by each of the first four jobs in π is at most $1 + \frac{(k-2.5)+k}{2} = k - \frac{1}{4}$. Since all jobs in the coalition benefit from the deviation, no pure NE in this setting is a 4-SE.

We now consider equilibria with mixed strategies. Clearly, the expected load on each machine has to be between $k-1$ and k. By Lemma A.1, on each machine there is at most one job that has a mixed strategy.

WLOG, assume M_1 is the most loaded machine. If there are 4 jobs that purely choose M_1 as their strategy, then the same deviation described for the pure case holds for these jobs. Otherwise, $k = 4$ and there are 3 jobs that purely choose M_1, and another job that has a mixed strategy and M_1 is in its support vector. WLOG, assume J_1 is the job on M_1 that has a mixed strategy and that M_2 is one of the other machines in its support. Let p denote the probability that J_1 chooses M_1. We also assume that the other jobs that choose M_1 are J_2, J_3 and J_4 (the expected load on M_1 is $3+p$).

Consider the following mixed deviation π of these jobs:

$$\pi_1 = \left(\frac{p}{2}, 1 - \frac{p}{2}, 0, 0, 0, 0, \ldots, 0\right)$$
$$\pi_2 = \left(\frac{1}{2}, 0, \frac{1}{2}, 0, 0, 0, \ldots, 0\right)$$
$$\pi_3 = \left(\frac{1}{2}, 0, 0, \frac{1}{2}, 0, 0, \ldots, 0\right)$$
$$\pi_4 = \left(\frac{1}{2}, 0, 0, 0, \frac{1}{2}, 0, \ldots, 0\right)$$

The strategies of the remaining jobs are unchanged. In the original joint strategy, the load observed by J_1 is 4, and the deviation decreases it to $1 + \left(\frac{p}{2} \cdot (3 - \frac{5}{2}) + \left(1 - \frac{p}{2}\right) 3\right) = 4 - \frac{3p}{4}$. As for the other jobs in the coalition, in the original joint strategy, the expected load observed by each job is $3+p$. In π, the expected load observed by each job is at most $1 + \frac{(1+p/2)+(3+p)}{2} = 3 + \frac{3p}{4}$. Since all jobs in the coalition benefit from the deviation, no mixed NE in this setting is a 4-SE. □

B Network Creation.

Theorem 4.3 *For any $\alpha \in (1,2)$, and any $n \geq 7$, there does not exist any 3-SE.*

Proof. We first establish the following sequence of lemmas.

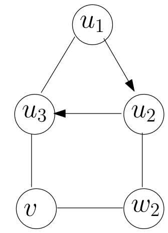


Figure 4: Sketch of the proof of Lemma B.3. By Lemma B.2, nodes w_2 and v must exist. By Lemma B.1, there must exist an edge between v and w_2. For $\alpha \in (1,2)$, it is beneficial for u_2 to remove the edge (u_2, u_3).

Lemma B.1. *For $\alpha \in (1,2)$, in any 3-SE, there does not exist any independent set of size 3 in $G(s)$, where s is a SE.*□

Lemma B.2. *For $\alpha \in (1,2)$, in any NE s, if there exists a set of nodes U that form a clique in $G(s)$, then if $u_1 \in U$ buys the edge to $u_2 \in U$, there must exists a node w_2 that is directly connected to u_2 but not to any other node $u \in U \setminus \{u_2\}$.*□

We use the above lemma to prove that a 3-SE cannot include triangles.

Lemma B.3. *For $\alpha \in (1,2)$, in any 3-SE s, there does not exist any triangle in $G(s)$.*□

Using the above lemmas, we derive a lower bound on the degree of each node in any 3-SE. Let $deg(v,G)$ be the degree of node v in the graph G.

Lemma B.4. *For $\alpha \in (1,2)$, in any 3-SE s, for every v, we have $deg(v, G(s)) \geq n-3$.*□

We now complete the proof of the theorem. By Lemma B.4, the degree of each node in any 3-SE must be at least $n-3$. Then, for $n \geq 7$, any edge removal can strictly decrease the cost of the node that bought it. Consider the edge (w,u). If w removes the edge, $B(w)$ decreases by $\alpha > 1$. We claim that $Dist(w)$ increases only by 1 (i.e., the only effect is that $\delta(w,u)$ increases from 1 to 2). To see this, note that for $n \geq 7$, if the degree of any node is at least $n-3$, then after removing (w,u), their degrees are at least $n-4$, and since for any $n \geq 7$, it holds that $n-4+n-4 > n-2$, they must have a common neighbor. In addition, for any node $u' \neq w, u$, both w and u' must have a common neighbor, since $n-4+n-3 > n-2$ (where $n-3$ and $n-4$ are the minimal respective degrees of u' and w). Therefore, by removing the edge (w,u), $Dist(w)$ increases by 1, while $B(w)$ decreases by $\alpha > 1$, so w strictly gains from the removal. □