Antisocial Behavior of Agents in Scheduling Mechanisms

Nandan Garg, Student Member, IEEE, Daniel Grosu, Member, IEEE, and Vipin Chaudhary, Member, IEEE

Abstract—Truthful task scheduling mechanisms are designed to cope with the selfishness of the participating agents. They assume that the agents are selfish; each agent’s goal is to maximize its own profit. However, this is not always the case; an agent may want to cause losses to the other agents besides maximizing its profit. Such an agent is said to be an antisocial agent. An antisocial agent will try to gain as much profit as possible relative to the other agents. This paper presents an antisocial strategy which can be used by the antisocial agents to inflict losses on the other agents participating in a task scheduling mechanism on related machines. This paper also studies, by simulation, the effect of different parameters, such as the degree of antisociality on the relative losses that can be inflicted on the participating agents.

Index Terms—Algorithmic mechanism design, antisocial agent, scheduling mechanism, simulation, Vickrey–Clarke–Groves (VCG) auction.

I. INTRODUCTION

Many current computer systems consist of geographically distributed resources coordinated by a central mechanism/protocol. The mechanism is responsible for allocating the load, controlling resources, disbursing remunerations, etc. The allocation and the disbursement are done according to certain private values (parameters) of the participating agents. Generally, resources are owned by self-interested organizations or agents with private goals. These agents may choose to reveal incomplete or untruthful information if that can deflect the outcome in their favor. An important problem in this setting is how to design protocols which obtain optimal outcomes even amid this selfish behavior. Mechanism design theory [1] (also known as implementation theory), which is a subfield of game theory, is an emerging approach to design such solutions for distributed multiagent optimization problems. The mechanism design approach uses incentives to motivate the agents to report their true parameters and follow the protocol. The mechanisms in which the agents maximize their utilities only by reporting their true parameters are called truthful (or strategy-proof) mechanisms.

In a mechanism, each participant has a privately known function called valuation which quantifies the agent’s benefit or loss. Payments are designed and used to motivate the participants to report their true valuations. The goal of each participant is to maximize the difference of its valuation and payment. An agent makes profit if the payment it received is greater than its true valuation, and it suffers losses if the payment it received is less than its true valuation. However, an agent can make profit and, at the same time, suffer relative losses (RLs) if the payment to that agent is reduced because some other agent misreports its valuation (as compared to the case in which all agents are reporting their true values).

Previous work in mechanism design assumed that the agents’ goal is to maximize their own profit and that an agent may deviate from the protocol if, by doing so, it can gain more profit [2], [3]. To tackle this problem, incentive compatible mechanisms were developed. It is implicit in the assumption previously stated that the agents do not care for others’ profits. In the real-world economy, sometimes, a company may accept a lower profit or even sell goods at loss if it is able to reduce the profit of its competitors. For a short period of time or in particular cases, such company may not give preference to maximizing its own profit. Such behavior is known as antisocial behavior, and the agents having such motivations are called antisocial agents. A related issue in the economic literature is referred to as “externalities” between bidders [4].Externalities refer to the preferences of a bidder specifying besides himself, who else he/she wants to be the winner. Here, we consider the antisocial agents who are interested in decreasing the profits of their rivals, whoever they may be. Such agents may exploit shortcomings in some mechanisms in order to inflict losses on the other agents. This is possible in mechanisms which employ payment schemes that depend on the reported valuations of the other agents (such as Vickrey payments). An antisocial agent needs to determine a bidding strategy that will allow it to inflict losses on the other participating agents while causing minimum losses to itself. In addition, given that the true value of an agent is a private value, the antisocial agent needs a way to infer the true values of the other agents such that it can modify its bid accordingly.

In this paper, we develop an antisocial strategy for the agents who intend to inflict losses on other participating agents. The strategy is presented in the context of online task scheduling on related machines [5]. The central mechanism allocates tasks of different sizes to different machines (agents) for execution and, then, issues payments as compensation for the use of their resources based on the Vickrey payment scheme. We study the effect of different agent’s parameters on the losses inflicted to the other agents. This antisocial strategy can also be applied to
other similar mechanisms in various contexts, which is part of our future research investigation.

A. Related Work

Recently, many researchers have used the mechanism design theory to solve problems in areas like resource allocation and task scheduling [6]–[8], congestion control, and routing [9], [10]. Nisan and Ronen [3] studied different mechanisms for shortest path and task scheduling. They proposed mechanisms to solve the shortest path problem and the problem of task scheduling on unrelated machines based on the popular Vickrey–Clarke–Groves (VCG) mechanism [11]–[13]. The VCG mechanisms can be applied only to problems where the objective functions are simply the sum of agent’s valuations and the set of outputs is finite. A general framework to design the truthful mechanisms for optimization problems where the agents’ private data are one real-valued parameter was proposed by Archer and Tardos [14]. Their framework can be applied to design mechanisms for optimization problems with general objective functions and restricted form of valuations. They also studied the frugality of shortest path mechanisms in [15]. A truthful mechanism that gives the overall optimal solution for the static load-balancing problem in distributed systems is proposed in [2]. Feigenbaum et al. [16] studied the computational aspects of mechanisms for cost sharing in multicast transmissions. A mechanism for low-cost routing in networks is proposed in [9]. A scenario where agents solve scheduling problems in a distributed manner is presented and analyzed in [17]. The results and the challenges of designing distributed mechanisms are surveyed in [18]. In [19], the authors proposed a market-driven strategy for auctions where agents change their behavior according to the market. Goldberg and Hartline [20] studied the problem of designing mechanisms such that no coalition of agents can increase the combined utility of the coalition by engaging in a collusive strategy. Johari [21] considered three different market models for resource allocation in communication networks and power systems and quantified the efficiency loss in these environments when the market participants are price anticipating. They also showed that, under reasonable assumptions, their mechanisms minimize the efficiency loss, when considering price-anticipating agents. The closest work to this paper is the work of Brandt and Weiß [22] in which they studied the behavior of antisocial agents in Vickrey auctions. They considered repeated Vickrey auctions in which the same item is auctioned in each round. They derived an antisocial strategy and introduced some notations to formalize the study of antisocial agents. This paper considers a different scenario where the tasks (items) are different but related in terms of their execution times.

B. Our Contributions

We consider a mechanism for online task scheduling on related machines and develop an antisocial strategy for the participating agents. We characterize the antisociality of an agent and analyze the effect of different degrees of antisociality on the losses that an agent can inflict on the other agents. We verify the correctness of the strategy by simulation. We also study the effect of changing different parameters on the amount of losses that an antisocial agent can inflict on the other participating agents.

C. Organization

In Section II, we present the formal model of the task scheduling problem and the scheduling mechanism. In Section III, we present a formal characterization of the antisocial behavior and, then, present the antisocial strategy for the scheduling mechanism. In Section IV, we present and discuss experimental results. Section V concludes this paper.

II. SCHEDULING PROBLEM AND MECHANISMS

A. Scheduling Problem

We consider here the problem of scheduling m independent tasks $T_1, T_2, \ldots, T_m$ on n machines $M_1, M_2, \ldots, M_n$. Each task $T_j$ is characterized by its processing requirement of $r_j$ units of time. Each machine $M_i$ is characterized by its processing speed $s_i > 0$, and thus, task $T_j$ would take $t^*_j = r_j/s_j$ time to be processed by $M_i$. A schedule $S$ is a partition of the set of tasks indexes into disjoint sets $S^i, i = 1, \ldots, n$. Partition $S^i$ contains the indexes corresponding to all the tasks allocated to $M_i$. The goal is to obtain a schedule $S$ minimizing a given objective function such as makespan, sum of completion times, etc. In the scheduling literature, this problem is known as scheduling on related machines [23].

B. Scheduling Mechanisms

In this section, we first describe the scheduling mechanism design problem and, then, present two scheduling mechanisms. The scheduling mechanisms are auction based, which means that the mechanism announces the tasks (size of tasks) to be executed and invites bids (the estimated execution time for the tasks) from machines. Based on the received bids, the mechanism allocates the tasks to the machines in the system. Generally, the machines deploy a daemon/process to bid on behalf of the machine. Such process providing service to machine $M_i$ is called an agent which is denoted by $A_i$. When a task is allocated to agent $A_i$ representing machine $M_i$, the agent submits the task to the machine. While the machine $M_i$ is busy in executing the task, agent $A_i$ may take part in the bidding process to gather more tasks for execution.

Definition 2.1: (Mechanism Design Problem). The problem of designing a scheduling mechanism is characterized by the following.

1) A finite set $S$ of allowed outputs. The output is a schedule $S(b) = (S^1(b), S^2(b), \ldots, S^n(b)) \in S$, which is computed according to the agents’ bids $b = (b_1, b_2, \ldots, b_n)$. Here, $b_i = (b_{i1}, b_{i2}, \ldots, b_{in})$ is the vector of values (bids) reported by agent $A_i$ to the mechanism.

2) Each agent $A_i$, ($i = 1, \ldots, n$), has, for each task $T_j$, a privately known parameter $t^*_{ji}$ ($j = 1, \ldots, m$) called the true value that represents the time required by agent $A_i$ to execute task $T_j$. The preferences of agent $A_i$ are given by a function called valuation $V_i(S(b), t_i) = \sum_{j \in S^i(b)} t^*_{ji}$ (i.e., the total time it takes to complete all tasks assigned to it). Here, $t_i = (t^*_{i1}, t^*_{i2}, \ldots, t^*_{im})$. 
3) Each agent’s goal is to maximize its utility. The utility of agent $A_i$ is $U_i(b, t) = P_i(b, t) - V_i(S(b), t_i)$, where $P_i$ is the payment handed by the mechanism to agent $A_i$.

4) The goal of the mechanism is to select a schedule $S$ that minimizes the makespan $C(S(b), t)$, (i.e., $\min(C(S(b), t))$, where $C(S(b), t) = \max_{j \in S_t(b)} t_j$.

An agent $A_i$ may report a value (bid) $b_j^i$ for a task $T_j$ that is different from its true value $t_j^i$. The true value characterizes the actual processing capacity of agent $A_i$. The goal of a truthful scheduling mechanism is to give incentives to agents such that it is beneficial for them to report their true values. Now, we give a formal description of a mechanism and define the concept of truthful mechanism.

**Definition 2.2: (Mechanism).** A mechanism is a pair of functions.

1) The allocation function: $S(b) = (S^1(b), S^2(b), \ldots, S^n(b))$. This function has, as an input, the vector of agents’ bids $b = (b_1, b_2, \ldots, b_n)$ and returns an output $S \in S$.

2) The payment function: $P(b, t) = (P_1(b, t), P_2(b, t), \ldots, P_n(b, t))$, where $P_i(b, t)$ is the payment handed by the mechanism to agent $A_i$.

**Definition 2.3: (Truthful Mechanism).** A mechanism is called truthful or strategy-proof if, for every agent $A_i$ with true value $t_i$, and for every bids $b \neq (b_1, \ldots, b_i, \ldots, b_n)$ of the other agents, the agent’s utility is maximized when it declares its true value $t_i$ (i.e., truth telling is a dominant strategy).

Nisan and Ronen [3] designed a truthful mechanism for a more general problem called scheduling on unrelated machines. In this case, the speed of machines depends on the task, i.e., for a task $T_j$ and agent $A_i$, the processing speed is $s_{ij}$. If $s_{ij} = s_i$ for all $i$’s and $j$’s, then the problem reduces to the problem of scheduling on related machines [23]. The authors provided an approximation mechanism called MinWork, minimizing the total amount of work. MinWork is a truthful mechanism. In the following, we present the MinWork mechanism.

**Definition 2.4: (Minwork Mechanism [3]).** The mechanism that gives an approximate solution to the scheduling problem is defined by the following two functions.

1) The allocation function: Each task is allocated to the agent who is able to execute the task in a minimum amount of time (allocation is random when there are more than one agent with a minimum type).

2) The payment function for agent $A_i$ is given by

$$P_i(b, t) = \sum_{j \in S^j_t(b)} \min_{j' \neq i} t_j'^i.$$  (1)

The MinWork mechanism can be viewed as running separate Vickrey auctions simultaneously for each task. Since an anti-social agent wants to infer the true value of another agent and bid in such a way that the other agent incurs losses, this is not possible if all tasks are auctioned simultaneously. In this paper, we consider a modification of MinWork mechanism that solves the problem of scheduling on related machines through the repeated Vickrey auctions with heterogeneous commodities. This mechanism can be viewed as running a sequence of separate Vickrey auctions, one for each task (of different size). The main difference from MinWork is that auctions are run in sequence and not simultaneously, and thus, an agent can apply the strategy in subsequent runs and inflict losses on other agents.

**Definition 2.5: [Modified MinWork Mechanism (MMW)].** The mechanism that gives an approximate solution to the problem of task scheduling on related machines is defined as follows.

For each task $T_j$ ($j = 1, 2, \ldots, m$), the two functions that define the mechanism are:

1) The allocation function: Task $T_j$ is allocated to the agent who is able to execute it in a minimum amount of time.

2) The payment function for agent $A_i$ is given by

$$P_i(b, t) = \min_{j' \neq i} t_j'^i, \quad j \in S^i(b).$$  (2)

The MMW solves the online scheduling problem [5], [24] where the jobs arrive to the scheduler over time and there is no a priori knowledge about the arrival of the next job. Online scheduling has many applications in different areas such as real-time systems [25], cluster computing [26], and wireless communications [27].

In the following, we present the protocol that implements the MMW.

**Protocol MMW:**

For each task $T_j$, $j = 1, \ldots, m$:

1) Agent $A_i$, $i = 1, \ldots, n$ submits bid $b_j^i$ to the mechanism.

2) After the mechanism collects all the bids, it does the following.
   1. Computes the allocation using the allocation function;
   2. Computes the payments $P_i$ for each agent $A_i$ using the payment function;
   3. Sends $P_i$ to each $A_i$.

3) Each agent receives its payment and evaluates its utility.

After receiving the payment, each agent evaluates its utility and decides on the bid values for the next task. This mechanism preserves the truthfulness property in each round.

III. ANTISOCIAL STRATEGY

In this section, we design an antisocial strategy for the agents participating in MMW. First, we present a formal characterization of agent’s antisocial behavior and, then, present the proposed antisocial strategy.

**A. Background**

The mechanisms presented in Section II are auction-based mechanisms which provide with efficient system-wide solutions even when the participating agents intend to deviate from the standard behavior in order to gain more profit. Such mechanisms are still vulnerable to manipulation by the agents who intend to inflict losses on the other agents rather than trying to maximize their own profit. An agent can exploit
the fact that certain payment schemes use the valuations of other agents to calculate the payment to the winner. One such popular scheme is the payment scheme used in the Vickrey auction [28]–[30], which is also known as second-price sealed-bid auction. In the Vickrey auction, the payment to the winner is equal to the bid of the second best agent for the auctioned item. As mentioned before, the MMW mechanism can be viewed as running a Vickrey auction separately for each task. Thus, an antisocial agent can reduce the profit of the winner (i.e., induce a loss) by bidding values close to the winner’s bid (assuming that the antisocial agent knows the bids of the winner).

Fig. 1 explains this antisocial behavior. In this figure, $v_1$, $v_2$, and $v_{as}$ are the true valuations of three different agents $A_1$, $A_2$, and $A_{as}$, respectively. When they all bid their true values, agent $A_1$ wins the auction and receives $v_2$ as payment. However, if agent $A_{as}$ wants to act as an antisocial agent, it can bid lower than its true value, such that its bid is between the bids of the best agent and the second best agent (in this case, $A_1$ and $A_2$). By carefully choosing this value, agent $A_{as}$ will be able to lower the profit of the best agent, as shown in Fig. 1. If the agent $A_{as}$ bids $b'_{as}$, then agent $A_1$ still wins the auction but receives a payment that is equal to $b'_{as}$ rather than $v_2$. Thus, effectively, the profit of $A_1$ is reduced by $v_2 - b'_{as}$.

The relative loss experienced here by the best agent is called RL. The RL can be defined as the loss experienced by an agent when the payment it receives is reduced as compared to the payment it would have received when all agents have reported their true valuations. This is different as compared to the absolute loss which is incurred when the payment is less than the agent’s true valuation. An agent may gain profit (the payment received is greater than its true valuation), but still incurring RLs, due to the antisocial behavior of another agent. The loss mentioned in the subsequent text is the RL, unless mentioned otherwise. Since the actual amount of loss varies depending on the size of the task, it is more meaningful to represent the RL as percentage.

RL is calculated as follows:

$$RL = \frac{P_T - P_i}{P_T} \times 100$$  \hspace{1cm} (3)

where $P_T$ is the payment received by the best agent $A_i$ when all agents, including $A_i$, report their true values, and $P_i$ is the payment received by the best agent $A_i$ when one of the agents is antisocial.

To formalize the study of antisocial behavior in the Vickrey auctions, Brandt and Weiß [22] introduced a parameter called the derogation rate of an agent. The derogation rate $d_i \in [0, 1]$ can be defined as the degree of antisocial behavior of an agent $A_i$. In other words, an agent will try to maximize the weighted difference between its utility and the utility of the other agents, where the weight is characterized by $d_i$. Thus, the derogation rate quantifies whether the agent gives preference to maximizing its profit or to inflicting losses to the other agents. The payoff of the agent depends on the derogation rate of the agent

$$\text{payoff}_i = (1 - d_i)U_i - d_i \sum_{i' \neq i} U_{i'}.$$  \hspace{1cm} (4)

Every agent tries to maximize its payoff. A regular agent has $d_i = 0$, and a purely destructive agent has $d_i = 1$. A balanced agent has $d_i = 0.5$, i.e., it gives equal weight to its utility and others’ losses.

The proposed strategy is for an agent participating in the MMW. As aforementioned, to be able to effectively inflict losses on the best agent, an antisocial agent needs to know the bids (true valuations) of the other agents. However, as stated previously, the valuations are private values that are not known to the other agents. By using the proposed strategy, an antisocial agent gains this information and uses it to inflict losses to the best agent. Brandt and Weiß [22] studied a similar problem but in the context of repeated Vickrey auctions, in which the same item is auctioned in each round. We base our strategy on those principles and develop a strategy for auctions involving tasks of different sizes. The strategy exploits the fact that the machines (agents) on which tasks are executed are related, which means that the true value of an agent (time required to execute a task) is proportional to the task size. We have also assumed that only the antisocial agent is manipulating its bids and that the other agents are reporting their true values.

In the proposed antisocial strategy, the antisocial agent bids according to its derogation rate and makes decisions based on whether it was allocated the last task or not. The agent starts by bidding its true valuation, and if it loses, it reduces its bid step by step so that it can gradually bid less than the bid of the best agent. The amount by which the antisocial agent $A_i$ reduces its bid in every step is characterized by the step down percent $q_i$. If $q_i$ is small, the agent reduces the bid by a very small amount and vice versa. When the antisocial agent’s bid is lower than the bid of the best agent, the antisocial agent wins and receives a payment equal to the bid of the best agent (which is assumed to be the true value of the best agent). After getting this information, the antisocial agent calculates its subsequent bids such that it can inflict an RL to the best agent. This calculation depends on the derogation rate of the agent, the true value of the best agent, and the task size. The antisocial agent keeps bidding accordingly for the subsequent tasks, thereby inflicting losses to the best agent.

B. Antisocial Strategy

In describing the proposed antisocial strategy, we use the following notations.
**Priceknown**

Boolean variable indicating whether $A_i$ knows the price paid for a task in the last round (it also indicates if $A_i$ won in the last round);

**PGreaterThanV**

Boolean variable indicating if the payment received in the last round was greater than the valuation;

**DecreaseBid**

Boolean variable to control execution; it indicates if the bid is to be decreased;

$t_j = t^i_j$

time required by antisocial agent $A_i$ to execute the current task $T_j$ (for simplicity, we denote $t^i_j$ by $t_j$ since the strategy is followed by agent $A_i$);

$t_{j-1} = t^i_{j-1}$

true value of agent $A_i$ for previous task $T_{j-1}$ ($i$ is implicit in $t_{j-1}$);

$P_i$

payment received by agent $A_i$ in the last round (if it won);

$b_{j-1} = b^i_{j-1}$

bid placed by agent $A_i$ for the previous task, which might be different from the current bid ($i$ is implicit in $b_{j-1}$);

$\epsilon$

an infinitesimal quantity;

$d_i$

derogation rate of the antisocial agent $A_i$;

$q_i$

step down percent of antisocial agent $A_i$’s bid.

The proposed antisocial strategy is shown in Fig. 2. Fig. 3 shows the state diagram representing various stages of the strategy and the associated transition criteria. For a particular task, an antisocial agent following this strategy can be in one of six stages. Depending on the current stage, other parameters, and the current allocation, the agent either stays in the same stage or moves to some other stage for the next task. Based on the current stage, the agent calculates the bid for the allocation of the current task. This strategy is followed only by the antisocial agents.

An antisocial agent $A_i$ starts in Stage 0 where it bids its true valuation. If the agent wins in Stage 0, the valuation of this agent is the lowest (and, thus, receives the payment which is equal to the true value of the second best agent). In this case, the best agent bids more than its true valuation to safeguard itself against possible attacks by other antisocial agents. If the presence of other antisocial agents is considered along with the best agent, the best agent will be able to safeguard only in some particular cases depending on its derogation rate and the derogation rate of the other antisocial agents. The right area (Stages 3, 4, and 5) corresponds to the case where an agent other than the best agent is antisocial, which might be generally the case.

An antisocial agent $A_i$ starts in Stage 0 where it bids its true valuation. If the agent wins in Stage 0, the valuation of this agent is the lowest (and, thus, receives the payment which is equal to the true value of the second best agent). In this case, the best agent transitions to Stage 1. We do not focus on this case because no loss can be inflicted to the best agent if the best agent itself is antisocial.

If the agent loses in Stage 0, then it transitions to Stage 3. In Stage 3, the antisocial agent reduces its bid by a small value anticipating to win. It keeps lowering the bid by a small percent $q_i$ in each run until it wins. When the agent wins, it receives a payment $P_i$ which is equal to the true value of the best agent. The transition is to Stage 4 or Stage 1 depending on whether the payment received $P_i$ is less than or greater than its true valuation $t_j$. If the agent wins and receives a payment less than its true valuation (i.e., the agent is not the agent with the lowest true value), then it bids according to the strategies of the right area and vice versa.

In Stage 4, the agent bids a value which is between its own true valuation and the true value of the best agent, thus inflicting a loss to the best agent. If the agent wins in Stage 4 and the payment it receives is less than its true valuation, it remains in the same stage; otherwise, it goes to Stage 1. If the agent loses in Stage 4, it moves to Stage 5 because it does not receive any payment (so the calculation of the bid cannot be done according to Stage 4). In Stage 5, the agent calculates the bid of the best agent using the information from the previous bid (intrinsic in the equation for Stage 5) and, then, bids accordingly to inflict losses to the best agent.
agent wins in Stage 5, it moves to Stage 4 or Stage 1 depending on whether the payment it receives is less than or greater than its own true valuation, respectively. If the agent loses in Stage 5, it remains in that stage where it can inflict losses on the best agent.

C. Effect on Makespan

As mentioned in Section III-B, the antisocial agent bids in such a way that, in Stage 3, it wins the auction and the task is allocated to it. Since the actual time in which the machine corresponding to the antisocial agent can execute the task is greater than the time required by the best agent, it increases the makespan of the system (as compared to the makespan obtained when no antisocial agent is present). The increase is equal to the difference in the execution time of the antisocial agent and the best agent. Assuming that the task $T_k$ is assigned to the antisocial agent $A_i$ in Stage 3 and that the best agent is denoted by $A_{\text{best}}$, the increase in makespan due to the antisocial strategy will be $t_{j_k} - t_{j_{\text{best}}}$. However, after this allocation, the antisocial agent moves to Stage 4 and, then, to Stage 5 where it will lose and it will not be allocated any further tasks. Thus, the action of the antisocial agent will increase the makespan, but the increase will be negligible in the long run (considering a large number of tasks).

IV. EXPERIMENTAL RESULTS

To analyze the proposed antisocial strategy, we developed a discrete event simulator and simulated the mechanism and the antisocial strategy. The simulation was run with different values for some of the parameters in order to study their effect on the losses that can be inflicted by an antisocial agent on the other participating agents. The simulation environment and the results are presented in this section. In the following, we use $\pi_i$ to denote the position of the antisocial agent $A_i$ in the sequence of agents sorted in an increasing order of their valuations. For example, $\pi_i = 1$ means the agent with the lowest true value, $\pi_i = 2$ means the agent with the second lowest true value, and so on.

A. Simulation Environment

The simulation was run with different combinations of parameters (presented in Table I) to study their effect on the agents’ losses. The parameters that were varied were the true values of the agents ($t_i$), the antisocial agent position ($\pi_i$), and the derogation rate ($d_i$). In the simulation, for a particular set of parameters, the mechanism allocates the tasks considering a normal scenario (no antisocial agent) and, then, considering the presence of one antisocial agent. The data related to the experiments (i.e., values of parameters, allocation, payments, etc.) in both cases are recorded.

We first simulated the strategy by considering a task scheduling scenario in which 16 agents are participating. To confirm the validity of the strategy when large number of agents are involved, we also simulated the strategy considering 128 agents. The results of the experiments involving 128 agents are presented in the second part of Section IV-B. The tasks are to be executed by the agents (resources), and the allocation is done according to the MMW mechanism presented in Definition 2.5. The value of $\epsilon$ (a chosen infinitesimal quantity) was set to...
0.01% of the task size (\( \epsilon \) is taken relative to the task size to remove scaling errors). In addition, the step down percent \( q \) was set to 5%. For each simulation experiment, one set of true values of the agents was generated according to the normal distribution. For a particular set of true values, the simulation was run for different antisocial positions \( (\pi_i = [1 \ldots 16]) \) (i.e., making the agent with the lowest true value as antisocial, then the second lowest as antisocial, and so on). For a particular set of true values and antisocial position, the derogation rates for the antisocial agent were varied from zero to one in steps of 0.1. For a particular set of true values, antisocial position, and derogation rate, the simulation was run for \( 10^5 \) tasks of different sizes. The task sizes were generated according to the uniform distribution. Since we are not proposing a new scheduling algorithm, the uniform distribution is sufficient to help us capture the economic aspect of the antisocial strategy. For this reason, we do not consider scheduling benchmarks. Moreover, as pointed out in [31], an analysis using randomly generated tasks is encouraged.

The recorded data are used to calculate the RL [as given in (3)], which the antisocial agent was able to inflict on the best agent. Since an antisocial agent at a particular antisocial position is able to inflict different amounts of RL to the winning agent at different derogation rates, we have recorded the maximum and the average RL percent that the agent is able to inflict (average is taken over varying derogation rates). Similarly at a higher level, for a particular set of bids, when agents at different positions are antisocial, they are able to inflict different amounts of losses to the winning agent, so we have recorded the maximum and average RL percent that an agent is able to inflict (averaged over different antisocial positions).

### B. Results

The following objectives were achieved by running the simulation.

1) The proposed antisocial strategy was experimentally validated.

2) The effect of derogation rate on the amount of inflicted loss was analyzed. The greater the derogation rate, the greater the loss inflicted by the antisocial agent.

3) The effect of the antisocial agent position on the amount of inflicted loss was analyzed. The smaller \( \pi_i (\pi_i \neq 1) \), the greater the average loss inflicted. The maximum loss is the same for all \( \pi_i \)'s (achieved at \( d = 1 \)).

4) The effect of the true value on the amount of inflicted loss was analyzed. The bigger the difference in the true values of the best agent and the second best agent, the bigger the RL to the best agent.

Fig. 4 shows the RL (as percent) inflicted by an antisocial agent. The figure shows the losses for the first 100 tasks only rather than the whole set of 10000 (since the RL percent becomes constant after the first few tasks). The agent starts by bidding its true valuation and decreases its bid (thus, increasing the RL to the best agent) until it is allocated the task and, thus, comes to know the valuation of the best agent (task 10). After this, the antisocial agent keeps bidding according to Stage 5, and it steadily keeps inflicting losses on the best agent for the rest of the tasks. It is notable here that the agent learns the true valuation of the best agent only after ten runs, which is quite negligible as compared to the number of tasks. The maximum loss inflicted is about 47% for the particular set of values of the parameters considered in Fig. 4.

In Fig. 5, we present the maximum RL inflicted to the best agent in two cases in which the agents at positions \( \pi_i = 2 \) and \( \pi_i = 4 \) are antisocial. It can be seen that if we consider the second agent as antisocial, then by reducing its bid, it is always able to inflict losses. The loss percent grows with the

### Table I

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range/Value</th>
<th>Parameters kept constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of agents (( n ))</td>
<td>16</td>
<td>-</td>
</tr>
<tr>
<td>( \epsilon )</td>
<td>0.01% of task size</td>
<td>-</td>
</tr>
<tr>
<td>Step down percent (( q_i ))</td>
<td>5%</td>
<td>-</td>
</tr>
<tr>
<td>True values (( t_j^i ))</td>
<td>normal(0 \ldots 2)</td>
<td>-</td>
</tr>
<tr>
<td>Antisocial position (( \pi_i ))</td>
<td>1 \ldots 16</td>
<td>( t_j^i ), ( \pi_i ), ( d_i )</td>
</tr>
<tr>
<td>Derogation rate (( d_i ))</td>
<td>0 \ldots 1</td>
<td>( t_j^i ), ( \pi_i ), ( d_i )</td>
</tr>
<tr>
<td>Task size (( \tau_j ))</td>
<td>uniform(0 \ldots 100)</td>
<td>( t_j^i ), ( \pi_i ), ( d_i )</td>
</tr>
<tr>
<td>Number of tasks (( \eta ))</td>
<td>10000</td>
<td>-</td>
</tr>
</tbody>
</table>
derogation rate of the agent. If an agent other than the second best agent is antisocial, it is able to inflict losses only when its bid is less than the second best agent’s bid; otherwise, it is not (even if it knows the valuation of the best agent). As can be seen from the figure, the antisocial agent in position 4 ($\pi_i = 4$) is able to inflict losses only after its derogation rate is 0.6. However, all the agents inflict maximum losses when their derogation rate is one, i.e., when they are purely destructive.

As it can be seen in Fig. 6, the maximum RL that any agent can inflict on the best agent is equal to the percent difference between the valuations of the best agent and the second best agent (in this case 54%). In addition, the average loss that an agent can inflict (with different derogation rates) decreases as the position of the antisocial agent increases (i.e., the higher the antisocial agent position, the lower the average loss). This is due to the fact that the antisocial agent is able to inflict losses only when its derogation rate is sufficiently high, thus making the bid of the antisocial agent less than the bid of the second best agent.

In the following, we present the results for a larger system which is composed of 128 agents. Fig. 7 shows that the pattern of losses inflicted on the best agent is the same even with the large number of agents. This is attributed to the following facts. We mentioned earlier in this section that there are three factors which determine the losses that an antisocial agent is able to cause to the best agent. They are the following: 1) the derogation rate; 2) the true values of the agents; and 3) the antisocial position $\pi_i$. At high derogation rates, the antisocial agent bids extremely close (just a little higher) to the winning agent, which inflicts a maximum loss to the winning agent. Thus, at high derogation rates and independent of the position $\pi_i$, the antisocial agent will be able to inflict losses to the winning agent. In addition, it is expected that, as the number of agents increases, there will be more agents that bid close to the bid of the winning agent. In this case, the agents can still inflict losses which could be as much as the difference in the bids of the best agent and the second best agent. This amount can be significantly high in highly heterogeneous environments. Moreover, in our problem, a large number of tasks are allocated to the best agent over time; thus, even if the loss inflicted by an agent seems to be less in one round, it will accumulate over time, and it will become significantly larger, fulfilling the aim of the antisocial agent. In Fig. 7, the average loss at higher antisocial positions is small because those agents are able to inflict losses only when their derogation rates are one. However, if they wish, they can keep their $d$ high to inflict significant losses up to the “maximum” value.

The increase in makespan due to the antisocial behavior was negligible. In a typical experiment, the increase in makespan was 0.612 time units, which was only 0.00088% of the total time (makespan).

V. CONCLUSION

As can be seen from the results presented in this paper, the presence of an antisocial agent in the task scheduling scenario can inflict losses on the other agents. This is due to the second price payment policy followed by the mechanism that we considered. The antisocial strategy presented in this paper can be applied by an agent who wants to inflict losses on the other agents. We analyzed the effect of different parameters on the antisocial strategy. In particular, agents with high derogation rates inflict more losses on the other agents. In addition, the amount of loss depends on the difference in true values of the agent with the lowest valuation, the agent with the second lowest valuation, and the antisocial agent (if different from above two). The amount of loss that can be inflicted also depends on the number of agents having the true values lying in between the true values of the best agent and the antisocial agent. In future research, we will consider the presence of more than one antisocial agent and study the effect of their strategies on the other agents’ profits. We also propose to investigate the antisocial behavior of agents in different mechanisms from other areas.

ACKNOWLEDGMENT

The authors would like thank the editor and the anonymous referees for their helpful and constructive suggestions, which considerably improved the quality of this paper.
REFERENCES


Nandan Garg

Daniel Grosu (S’99–M’03) received the B.S. degree in engineering (automatic control and industrial informatics) from the Technical University of Iasi, Iasi, Romania, in 1994, and the M.Sc. and Ph.D. degrees in computer science from The University of Texas at San Antonio, San Antonio, in 2002 and 2005, respectively. Currently, he is an Assistant Professor with the Department of Computer Science, Wayne State University, Detroit, MI. His research interests include incentive-based computing, game theory, mechanism design, distributed systems, security, and parallel processing.

Vipin Chaudhary (S’89–M’92) received the B.Tech. (Hons.) degree in computer science and engineering from the Indian Institute of Technology, Kharagpur, in 1986. He received the M.S. degree in computer science and the Ph.D. degree in electrical and computer engineering from The University of Texas at Austin, Austin, in 1989 and 1992, respectively.

In 2000, he was with Corio, Inc., San Carlos, CA, in various capacities, finally as a Chief Architect. From January 1992 to May 1992, he was a Postdoc- toral Fellow with the Computer and Vision Research Center, The University of Texas at Austin. He is currently an Associate Professor of electrical and computer engineering and computer science with the Department of Computer Science, Wayne State University, Detroit, MI, and he is the Associate Direc- tor of the Institute for Scientific Computing, Wayne State University. He is also the Senior Director for Advanced Development at Cradle Technologies, Inc., Mountain View, CA. His current research interests are programming environments for high-performance computing systems, scientific computing, computer-assisted surgery and medical image processing, sensor networks, and bioinformatics. He has published more than 100 peer-reviewed papers in the broad areas of parallel and distributed computing, image processing, security, and scientific computing.

Dr. Chaudhary has served on program committees for numerous international conferences in the aforementioned areas. He is on technical advisory boards for several companies. He received the Research Initiation Award from the National Science Foundation in 1993, and his research has been funded by federal and state agencies and industry. He received the President of India Gold Medal for academic excellence from the Indian Institute of Technology.