

Mechanism Design for Strategic Project Scheduling

Pradeep Varakantham and Na Fu

School of Information Systems, Singapore Management University
pradeepv@smu.edu.sg, nafu@smu.edu.sg

Abstract

Organizing large scale projects (e.g., Conferences, IT Shows, F1 race) requires precise scheduling of multiple dependent tasks on common resources where multiple selfish entities are competing to execute the individual tasks. In this paper, we consider a well studied and rich scheduling model referred to as RCPSP (Resource Constrained Project Scheduling Problem). The key change to this model that we consider in this paper is the presence of selfish entities competing to perform individual tasks with the aim of maximizing their own utility. Due to the selfish entities in play, the goal of the scheduling problem is no longer only to minimize makespan for the entire project, but rather, to maximize social welfare while ensuring incentive compatibility and economic efficiency. We show that traditional VCG mechanism is not incentive compatible in this context and hence we provide two new practical mechanisms that extend on VCG. These new mechanisms referred to as Individual Completion based Payments (ICP) and Social Completion based Payments (SCP) provide strong theoretical properties including strategy proofness.

1 Introduction

Resource Constrained Project Scheduling Problems (RCPSPs) [Kolisch and Sprecher, 1996; Pinedo, 2008] have been studied extensively in the context of manufacturing, project management and logistics. The focus is on minimizing the time taken to complete a set of tasks pertaining to a common project within the context of resource constraints. Given the NP-Hard complexity of solving RCPSPs, existing research has focussed on methods for generating high quality solutions efficiently [Vanhoucke and Coelho, 2016; Varakantham *et al.*, 2016; Schutt *et al.*, 2013; Fu *et al.*, 2012].

While RCPSPs represent task scheduling, they do not represent allocation and execution of tasks when multiple selfish individuals/companies are present. For instance, consider the project of organizing an F1 race in a city (can be extended to any other major event or project). There are multiple tasks (putting up barricades, organizing logistics, selling tickets etc.) each of which can be done by different individuals or

companies, each having their selfish interest (typically maximizing profits) in performing the task. Resources on which tasks will be performed are the roads, locations for congregating crowds, etc. that are typically controlled by the city. In order to represent such problems, we focus on a strategic variant of RCPSP in this paper.

Apart from the underlying RCPSP, we have the following additions in a strategic RCPSP: (i) There are multiple selfish agents interested in performing tasks. More importantly, different agents can propose different durations and costs for attending a task. Such information is not publicly available at the time of task bidding; (ii) For the central authority, there is a value associated with finishing the project at a certain time. This is typically a monotonically decreasing function of project makespan (duration to finish the project) that rewards early finish to the project. Given these two additions, our goal is to design a mechanism that will ensure agents truthfully reveal their types (durations and costs for tasks) and also finish the project at the earliest possible time. Truthful revelation is a desirable property to avoid delays in projects (due to wrong reporting by agents) or unhappiness on part of agents executing tasks (due to payments that are lower than desired).

There has been a significant focus on mechanism design for machine scheduling problems [Heidenreich *et al.*, 2007]. The general VCG mechanism [Vickrey, 1961; Clarke, 1971; Groves, 1973] can be applied for certain basic scheduling problems without precedence or resource constraints among tasks. A characterisation of mechanisms for two machines showing that only task-independent mechanisms can be truthful was provided in [Dobzinski and Sundararajan, 2008]. Nisan and Ronen [Nisan and Ronen, 2001] examined lower and upper bounds on approximation using deterministic and randomized mechanisms for the unrelated machines scheduling problem. There are multiple key differences between work on selfish machine scheduling and the work presented in this paper: (i) In selfish scheduling, agents are competing for resources to finish their own job. However, in strategic RCPSP, agents are competing to do tasks that are part of a project. There is no competition for resources and are allocated by central authority; (ii) There are temporal dependencies between tasks in RCPSP; and (iii) Each task can require multiple resources of different types and there is a capacity for each resource type in RCPSP. In selfish scheduling, there is typically just one type of resources.

Another thread of interest with respect to mechanism design is in task allocation settings. In [Archer and Tardos, 2007], the problem of hiring a team of agents to perform a project was considered and the cost of truthfulness was evaluated. As an extension of that problem, scenarios where the agents first form teams and then bid for as consolidated groups rather than as individuals were examined in [Skowron *et al.*, 2017]. Different from our work, the problem of hiring a team is assumed without task dependencies and resource capacity constraints. Porter *et al.* [Porter *et al.*, 2008] considered task allocations where tasks can fail. They provided mechanisms where payment function is dependent on outcome of task execution and not just on reported types. We build on this idea in our mechanisms described later. There have been other works on mechanism design for machine scheduling problems, however there is a major difference. In RCPSP, the processing of tasks might need multiple resources of different types, rather than one machine. Therefore, the focus here is on developing scalable mechanisms for a more general and practically relevant project scheduling problem.

There has also been existing research on resource unconstrained scheduling problems in a decentralised manner using game theory. Nash equilibria for scheduling with controllable processing times was examined in [Agnētis *et al.*, 2015] and in [Estévez-Fernández, 2012], an approach for penalty and reward sharing in projects was proposed. Our work focusses on problems with resource capacity constraints and fixed processing times.

We make three key contributions with respect to designing mechanisms for the strategic RCPSP: (i) We formally define the strategic model and its objective. (ii) We then show that the well known VCG mechanism is not incentive compatible for the strategic RCPSP. (iii) Finally, we provide two mechanisms that build on VCG, namely SCP and ICP with strong theoretical properties. Specifically, our ICP mechanism is able to ensure that truthful behaviour is the dominant strategy and also that it is better for agents to participate in the project with their true type. One practical difference between SCP and ICP is the timing of making payment to agents: in SCP, payments are provided after the whole project completes; while in ICP, an agent can get the payment immediately after the allocated task(s) are performed.

2 Background: RCPSP

The Resource Constrained Project Scheduling Problem (RCPSP) consists of a set of tasks $\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_N\}$. Each task τ_q has a duration denoted by d_q where $q = 1, \dots, N$. There are K types of reusable resources each with a limited capacity represented by C_k . Each task τ_q requires r_{qk} units of type k resources during execution where $k = 1, \dots, K$. In addition, two dummy tasks τ_0 and τ_{N+1} with zero durations are introduced to represent project source and sink, respectively.

A *schedule* is an assignment of start times to all tasks, i.e. a vector (s_1, s_2, \dots, s_N) , where s_q represents the start time of τ_q . Let e_q be the end time of task τ_q , we then have $s_q + d_q = e_q$. The project *makespan*, which is the start time of the sink activity τ_{N+1} , can be given by $\mathcal{M} = s_{N+1} = \max_{q=1, \dots, N} e_q$.

There are two types of scheduling constraints, namely

precedence constraints and *resource constraints*. Precedence constraints specify precedence relationships between tasks. τ_q precedes τ_z implies that τ_z cannot start before τ_q ends, i.e., $s_z - s_q \geq d_q$. Resource constraints restrict the total resource consumption during project execution. A schedule is resource feasible if at each time t , the total demand of tasks for any resource type k does not exceed its capacity C_k .

Typically, the objective of RCPSP is to find a start time schedule with the minimum makespan that satisfies both precedence and resource constraints. Instead of a start time schedule, in this paper we focus on computing an execution policy referred to as Partial Order Schedule (POS) ([Policella *et al.*, 2007]) that minimizes the makespan. A POS represents a set of partially ordered tasks such that any embedded temporal feasible solution is guaranteed to be resource feasible. One property of POS is that it can provide an online policy in terms of when and what tasks to start. Within a POS, each task retains a set of feasible start times thereby providing more flexibility than a traditional schedule, where each task is restricted to start at a specific time.

3 Model: Strategic RCPSP

We consider a strategic model of RCPSP where there are multiple selfish agents bidding to perform individual tasks. Unlike in RCPSP, where the only goal is to compute an execution strategy or start time schedule, in a strategic RCPSP, there are three specific goals: (i) An execution strategy (POS) for tasks to be executed; (ii) Allocation of tasks to agents; (iii) Payments to agents for performing allocated tasks. In terms of representation, there are three key components: the underlying RCPSP, valuation function associated with project completion, and models for selfish agents interested in performing individual tasks.

First, we have the underlying RCPSP model as introduced in Section 2. There are precedence constraints between tasks of the form $s_z - e_q \geq T_{qz}$, where T_{qz} is a non-negative value that specifies τ_z can only be started when τ_q has already finished for a certain time period of T_{qz} . The second component that adds to the strategic nature of RCPSP is a valuation, $V(\cdot)$ for completion of the project, which is typically a monotonically non-increasing function over the value of project makespan. Finally, we have a set of agents, $A = \{a^1, \dots, a^M\}$ competing for performing tasks. The set of tasks of interest for agent i is given by Q^i . Let $d_q^i(\theta^i)$ and $c_q^i(\theta^i)$ represent the duration of task q and the cost for executing that task, respectively. The actual type of i is denoted by $\theta^i = (\mathbf{c}^i, \mathbf{d}^i)$, where \mathbf{c}^i and \mathbf{d}^i are vectors of costs and durations with

$$\mathbf{c}^i = \{c_q^i(\theta^i) | q \in Q^i\} \quad \mathbf{d}^i = \{d_q^i(\theta^i) | q \in Q^i\}.$$

However, the actual type is a private information when agents express their interest in executing tasks and agents may make false declarations for self interests.

We aim to solve the strategic RCPSP by designing mechanisms that can provide incentives to agents so as to reveal truth about their types. This will ensure that individual tasks are executed as expected by agents, which in turn guarantees a successful project delivery. We wish to avoid situations where individual agents indicate lower durations or lower costs to

increase the chance of executing a task (in the hope of gaining revenue from the task and not executing it on time). Given that it is hard to predict task durations accurately even for individual agents, mechanisms that over penalize (e.g., million dollars for any wrong reporting) after realising true durations are not applicable.

4 Mechanisms

In this work, we focus on *direct revelation* mechanisms. That is, the mechanism is directly based on the type reported by agents. Here is the flow of various steps involved in getting a project executed: (1) Agents report types. (2) Based on agents' reported types, a central authority decides on the task allocation and execution policy. (3) Agents then perform their allocated tasks according to the execution policy. (4) Depending on the mechanism, they can receive payments/contracts based either on reported types or on actual type (that is available after actual completion of tasks).

Let Θ^i denote the type space for agent i . Agent i 's reported type $\hat{\theta}^i$ may be different from its true type θ^i in order to gain a favorable outcome. A mechanism, Γ is defined as:

$$\Gamma = (\Theta, g(\cdot))$$

where $\Theta = \Theta^1 \times \dots \times \Theta^M$. The function $g(\hat{\theta})$ maps the declaration of agents, $\hat{\theta}$ to an output $o \in O$ that is defined as:

$$(\mathbf{f}(\hat{\theta}), \mathbf{y}(\hat{\theta}), \mathbf{p}(\hat{\theta}))$$

where $\mathbf{f}(\hat{\theta})$ is the allocation vector with $f_q^i(\hat{\theta}) = 1$ indicating that task τ_q is allocated to agent i and $f_q^i(\hat{\theta}) = 0$ otherwise. $\mathbf{y}(\hat{\theta})$ denotes the partially ordered schedule, with $y_{qz}(\hat{\theta}) = 1$ representing that task τ_z can only start after task τ_q finishes. The execution policy \mathbf{y} can also store information about the exact precedence lags, which is the minimum time to wait after previous task finishes. Finally, $\mathbf{p}(\hat{\theta})$ represents the payment to individual agents from the central for executing the allocated tasks. If a task is not allocated to an agent, then both cost and duration for that agent on the task will be zero.

One key differentiating factor in this mechanism when compared to mechanisms for task allocation or traditional machine scheduling problems is the presence of the execution policy, POS. When agents provide false reports of processing times, the deterministic schedule with fixed starting times of tasks cannot be employed to evaluate the impact of their declarations, because any wrong reporting of durations can render the schedule infeasible. However, by using a POS, we are guaranteed to get a feasible schedule and consequently values of false reporting can be evaluated. We provide specifics of the mechanisms employed in the next two subsections.

4.1 Payments and Participant Utilities

We first define some of the basic terms required for computing payments and utilities. Given the allocation \mathbf{f} , POS \mathbf{y} and revealed type $\hat{\theta}$, the project makespan can be given by $\mathcal{M}(\mathbf{f}(\hat{\theta}), \mathbf{y}(\hat{\theta}), \hat{\theta}) = s_{N+1}(\mathbf{f}(\hat{\theta}), \mathbf{y}(\hat{\theta}), \hat{\theta})$ with $N+1$ referring to the sink task. Value for the central authority is denoted by

$V(\mathcal{M}(\mathbf{f}(\hat{\theta}), \mathbf{y}(\hat{\theta}), \hat{\theta}))$, where $V(\cdot)$ is a monotonically non-increasing function over the value of project makespan.

Given the revealed type $\hat{\theta}$, welfare for all participants denoted by $\mathcal{W}(\mathbf{f}(\hat{\theta}), \mathbf{y}(\hat{\theta}), \hat{\theta})$, can be defined as the center's value minus the cost incurred by all agents, i.e.,

$$\mathcal{W}(\mathbf{f}(\hat{\theta}), \mathbf{y}(\hat{\theta}), \hat{\theta}) = V(\mathcal{M}(\mathbf{f}(\hat{\theta}), \mathbf{y}(\hat{\theta}), \hat{\theta})) - \sum_i c^i(\hat{\theta})$$

where

$$c^i(\hat{\theta}) = \sum_{q \in Q_i} C_q^i(\hat{\theta}), \quad C_q^i(\hat{\theta}) = \begin{cases} c_q^i(\hat{\theta}^i) & \text{if } f_q^i(\hat{\theta}) = 1 \\ 0 & \text{otherwise} \end{cases}$$

with $C_q^i(\hat{\theta})$ representing the cost for agent i by performing τ_q (if it is allocated to i), and $V = \sum_i V^i$ with V^i representing the value contributed by agent i . The solution framework we provide allow for a flexible design of the individual valuation V^i . For example, if $V(\cdot)$ is linear with the sole makespan, a direct way to evaluate an agent's contribution by performing task τ_q is by determining a ratio $0 \leq \rho_{iq} \leq 1$ which can be calculated as the cumulative portions of the task duration over the total makespan, divided by the number of tasks in parallel. That is, $V^i = V \sum_q \rho_{iq} = V \sum_q \sum_t 1/(|T_t| \cdot \mathcal{M})$, where the duration of τ_q can be divided into d_q^i time units and $|T_t|$ is the number of tasks in parallel at the t^{th} time unit ($t = 1, \dots, d_q^i$). Note that properties of our proposed mechanisms (which will be introduced in later sections) hold independent of the explicit representation of individual valuation.

Let the superscript $-i$ on a vector denote that the term for agent i has been omitted from the vector. For example, $\theta^{-i} = (\theta^1, \dots, \theta^{i-1}, \theta^{i+1}, \dots, \theta^M)$. $\mathcal{W}^{-i}(\cdot)$ is employed to indicate the welfare of all participants excluding i . Each agent would receive a certain amount of payment from the center after performing the allocated task(s). In the following, we will present payments corresponding to three different mechanisms: (i) VCG payment denoted by p_{VCG}^i . (ii) Social Completion based Payment denoted by p_{SCP}^i . (iii) Individual Completion based Payment denoted by p_{ICP}^i .

VCG Mechanism: Following VCG mechanism, payment for i can be defined as the difference of social welfare of other agents (denoted by \mathcal{W}_{VCG}^{-i}) when i is present, and the optimal welfare (denoted by \mathcal{W}_{VCG}^{-i}) when i is not present. That is,

$$p_{VCG}^i(\hat{\theta}) = \mathcal{W}_{VCG}^{-i}(\mathbf{f}(\hat{\theta}), \mathbf{y}(\hat{\theta}), \hat{\theta}) - \mathcal{W}_{VCG}^{-i}(\mathbf{f}^{-i}(\hat{\theta}^{-i}), \mathbf{y}^{-i}(\hat{\theta}^{-i}), \hat{\theta}^{-i}) \quad (1)$$

It should be noted that with VCG, agents would have incentive to increase the payoff by misreporting their types, since the utility function depends on the reported types.

SCP Mechanism: Motivated by VCG's lack of incentive compatibility due to dependence on reported types, we propose a payment function that depends not only on the task allocation and execution policy (derived based on the declarations $\hat{\theta}$), but also on the *actual* types (denoted by θ) of

agents at completion, which can be defined as:

$$p_{SCP}^i(\hat{\theta}) = \mathcal{W}^{-i,i}(\mathbf{f}(\hat{\theta}), \mathbf{y}(\hat{\theta}), \theta) - \mathcal{W}^{-i}(\mathbf{f}(\hat{\theta}^{-i}), \mathbf{y}(\hat{\theta}^{-i}), \hat{\theta}^{-i}) \quad (2)$$

where $\mathcal{W}^{-i,i}$ is a sum of welfare for other participants when i is present and the valuation of i (denoted by V^i), i.e.,

$$\mathcal{W}^{-i,i}(\mathbf{f}(\hat{\theta}), \mathbf{y}(\hat{\theta}), \theta) = \mathcal{W}^{-i}(\mathbf{f}(\hat{\theta}), \mathbf{y}(\hat{\theta}), \theta) + V^i \quad (3)$$

Note that $\mathcal{W}^{-i,i}$ is calculated corresponding to the allocation and POS derived based on declarations $\hat{\theta}$, but with durations and costs taken from the real types. Since the truth is not publicly known, this term can only be calculated after execution of tasks and realisation of costs and durations. The second term \mathcal{W}^{-i} of Eqn 2 on the other hand computes the welfare for all agents excluding i when i is not present, according to reported types of all other agents excluding i , i.e., $\hat{\theta}^{-i}$.

In the first term of payment definition $p^i(\hat{\theta})$ in Eqn 2, the third attribute of welfare depends on the real types of all agents. In other words, agent i has to wait until all tasks by other agents have been completed to receive the payment.

ICP Mechanism: Motivated by the potential inefficiency due to the payment timing, the ICP mechanism relies on a payment function, p_{ICP}^i that agents can get payments immediately after the individual allocations are processed, i.e.

$$p_{ICP}^i(\hat{\theta}) = \mathcal{W}^{-i,i}(\mathbf{f}(\hat{\theta}), \mathbf{y}(\hat{\theta}), (\theta^i, \hat{\theta}^{-i})) - \mathcal{W}^{-i}(\mathbf{f}(\hat{\theta}^{-i}), \mathbf{y}(\hat{\theta}^{-i}), \hat{\theta}^{-i}) \quad (4)$$

where

$$\mathcal{W}^{-i,i} = \mathcal{W}^{-i}(\mathbf{f}(\hat{\theta}), \mathbf{y}(\hat{\theta}), (\theta^i, \hat{\theta}^{-i})) + V^i \quad (5)$$

Unlike in Eqn 3, $\mathcal{W}^{-i,i}$ in Eqn 5 can be calculated corresponding to the allocation and POS from declarations $\hat{\theta}$, and the real type of agent i only. Under ICP mechanism, the payment can be decided immediately after an individual agent completes the allocated tasks and contracting can take place before the entire project completion.

Irrespective of the three payment rules, utility for individual agent can be defined as the payment received minus the cost incurred in executing the allocated tasks. That is,

$$u^i(\hat{\theta}) = p^i(\hat{\theta}) - C^i(\hat{\theta}) \quad (6)$$

Utility of the centre is the difference between the center's value and the total payments made to all agents:

$$u^\#(\hat{\theta}) = V(\mathcal{M}(\mathbf{f}(\hat{\theta}), \mathbf{y}(\hat{\theta}), \theta)) - \sum_i p^i(\hat{\theta}) \quad (7)$$

where the second term can be defined by Eqn 2 or 4 based on which mechanism is applied.

4.2 VCG and SCP

It is well known that if the utility of a participant is depending other participants' types (because of task duration in the context of strategic RCPSP), such interdependent valuations

are not incentive compatible to truthfully reveal the type for VCG mechanism ([Postlewaite and McLean, 2015]). We will use a concrete example to show that VCG is not incentive compatible, while SCP is.

Consider a simple case with only one task and no cost involved¹. Let $\hat{\theta}^i$ denote the reported task duration from agent i . After the centre receives all declarations, an allocation together with a payment are calculated. Let $\hat{\theta}_1$ and $\hat{\theta}_2$ represent the lowest and second lowest reported durations, respectively. As in most real world cases, we suppose centre's value, $V(\cdot)$ is a non-increasing function with the task duration and i is the winning agent, i.e., $\hat{\theta}^i = \hat{\theta}_1 < \hat{\theta}_2$.

According to VCG payment, an agent would be paid based on the value increase of other agents due to its presence and the total value with and without i are $V(\hat{\theta}_1)$ and $V(\hat{\theta}_2)$, respectively. Thus, the utility of i can be presented as $u_{VCG}^i = V(\hat{\theta}_1) - V(\hat{\theta}_2)$ and utilities of all agents except the winner would be zero. Such a mechanism is not incentive compatible, because it is solely based on declaration of each agent and the agents can report wrongly to derive higher incentive.

We overcome this barrier by designing a payment function which is not only based on the declarations, but also on the actual execution of tasks. Following our SCP mechanism, the utility of the winner i can be instead represented as, $u_{SCP}^i = V(\theta_1) - V(\hat{\theta}_2)$, where θ_1 is the actual value of the winner i .

There are three possible declarations: truth-telling, over-reporting and under-reporting. We show that under SCP, agents have no incentive to misreport for each scenario.

- $\hat{\theta}_1 < \hat{\theta}_2 < \theta_1$. In this case, i wins by over-reporting. Given $V(\cdot)$ is non-increasing, we then have $u_{SCP}^i < 0$ (utility is negative).
- $\hat{\theta}_1 < \theta_1 < \hat{\theta}_2$. Since $\theta_1 < \hat{\theta}_2$, by reporting the truth, i can secure the task. There is no incentive to report a lower value as no utility gain can be obtained.
- $\theta_1 < \hat{\theta}_1 < \hat{\theta}_2$. Similar to the previous case, besides receiving no utility gain, there is also a potential risk to losing the task by reporting a higher value. Thus, i has no incentive to lie.
- $\theta_1 = \hat{\theta}_1 < \hat{\theta}_2$. Under this scenario, i secures the task and gets a positive utility given by $V(\theta_1) - V(\hat{\theta}_2)$.

In other words, truth-telling is dominant in this example under the SCP mechanism. We show properties of SCP and ICP with respect to incentive compatibility in later sections.

4.3 MILP for Strategic RCPSP

Another key feature of our proposed SCP and ICP is the integration of an execution policy (i.e., POS) together with task allocation, so that an online decision is always available to tell at each time, what tasks to be executed on which resources. Note that online execution of the generated POS is beyond the scope of this paper. Overall, the proposed solution framework for solving strategic RCPSP contains three processes: task allocating, task executing and payment making.

¹Note that even if cost is not present, time can always be treated as an opportunity cost. Pro-bono projects are one example of cases where cost is irrelevant.

Input: $\mathbf{r}, \mathbf{C}, \mathbf{T}, \hat{\theta}$

Output: \mathbf{f}, \mathbf{y}

$$\max_{\mathbf{f}, \mathbf{y}} V(s_{N+1}) - \sum_i C^i(\hat{\theta})$$

$$C^i(\hat{\theta}) = \sum_{q \in Q_i} C_q^i(\hat{\theta}) \quad \forall i \quad (8)$$

$$C_q^i(\hat{\theta}) = c_q^i(\hat{\theta}^i) \cdot f_q^i(\hat{\theta}) \quad \forall q, i \quad (9)$$

$$d_q = \sum_{a_i \in A} f_q^i(\hat{\theta}) \cdot d_q^i(\hat{\theta}^i) \quad \forall q \quad (10)$$

$$\sum_{a_i \in A} f_q^i(\hat{\theta}) = 1 \quad \forall q \quad (11)$$

$$e_q = s_q + d_q \quad \forall q \quad (12)$$

$$s_z \geq e_q + T_{qz} - M \cdot (1 - y_{qz}) \quad \forall q, z \quad (13)$$

$$x_{q,z}^k \leq \min\{r_{qk}, r_{zk}\} \cdot y_{qz} \quad \forall q, z, k \quad (14)$$

$$\sum_z x_{q,z}^k = \sum_z x_{z,q}^k = r_{qk} \quad \forall q \neq 0, N+1, k \quad (15)$$

$$\sum_z x_{0,z}^k = \sum_z x_{z,N+1}^k \leq C_k \quad \forall k \quad (16)$$

$$y_{qz} \in \{0, 1\} \quad \forall q, z \quad (17)$$

$$y_{qz} + y_{zq} \leq 1 \quad \forall q, z \quad (18)$$

Table 1: ALLOCPOS()

We build on the idea of generating POSs for RCPSP/max using a flow-based continuous time linear model in [Fu *et al.*, 2016; Varakantham *et al.*, 2016] and provide an MILP formulation in Table 1 to determine the optimal allocation and POS. Given the declarations from agents, temporal and resource constraints among tasks, the objective of the MILP is to generate the best task allocation and execution policy POS, where 'best' here is characterized with respect to maximizing welfare for all agents based on the revealed types.

Eqns 8 - 9 compute the actual cost of agent i by processing τ_q and if it is allocated to i by allocation f . The duration of τ_q is represented by Eqn 10. Eqn 11 guarantees that for each τ_q , only one agent is assigned to it. Eqns 12 - 18 are for POS generation. The resource flow variable $x_{q,z}^k$ represents the number of resource k transferred directly from τ_q to τ_z . The sequencing variables $y_{q,z}$ are for POS construction. Note that in the MILP, start time variables are only used for policy computing. Eqn 13 links starting times of τ_q and τ_z with y_{qz} . It is active when $y_{qz} = 1$ which enforces the precedence relationship. If τ_q precedes τ_z , the maximum resource flow from τ_q to τ_z is forced to be $\min\{r_{qk}, r_{zk}\}$, as in Eqn 14. Eqns 15 - 16 are flow conservation constraints. Eqn 18 covers the three relationships between two tasks, either one task precedes another, or both being executed in parallel. All constraints in the optimization model are linear. If V is linear or quadratic, the model can be solved using solvers like CPLEX. In fact, as indicated in [Fu *et al.*, 2016], problems with up to 30 activities can be executed efficiently when minimizing makespan.

θ^i	Actual type of agent i
$\hat{\theta}^i$	Reported type of agent i
θ^{-i}	Actual types of agents except i , $(\theta^1 \dots \theta^{i-1}, \theta^{i+1} \dots \theta^M)$
θ	(θ^i, θ^{-i})
$\hat{\theta}$	$(\hat{\theta}^i, \hat{\theta}^{-i})$
$\bar{\theta}$	$(\theta^i, \hat{\theta}^{-i})$
$\tilde{\theta}$	$(\hat{\theta}^i, \theta^{-i})$
$\mathbf{f}^*, \mathbf{y}^*$	$\arg \max_{\mathbf{f}, \mathbf{y}} \mathcal{W}(\mathbf{f}(\theta), \mathbf{y}(\theta), \theta)$
$\mathbf{f}_{-i}^*, \mathbf{y}_{-i}^*$	$\arg \max_{\mathbf{f}, \mathbf{y}} \mathcal{W}^{-i}(\mathbf{f}(\theta^{-i}), \mathbf{y}(\theta^{-i}), \theta^{-i})$
$\hat{\mathbf{f}}_{-i}^*, \hat{\mathbf{y}}_{-i}^*$	$\arg \max_{\mathbf{f}, \mathbf{y}} \mathcal{W}^{-i}(\mathbf{f}(\hat{\theta}^{-i}), \mathbf{y}(\hat{\theta}^{-i}), (\hat{\theta}^{-i}))$
$\hat{\mathbf{f}}^*, \hat{\mathbf{y}}^*$	$\arg \max_{\mathbf{f}, \mathbf{y}} \mathcal{W}(\mathbf{f}(\hat{\theta}), \mathbf{y}(\hat{\theta}), (\hat{\theta}))$
$\bar{\mathbf{f}}^*, \bar{\mathbf{y}}^*$	$\arg \max_{\mathbf{f}, \mathbf{y}} \mathcal{W}(\mathbf{f}(\bar{\theta}), \mathbf{y}(\bar{\theta}), (\bar{\theta}))$
$\tilde{\mathbf{f}}^*, \tilde{\mathbf{y}}^*$	$\arg \max_{\mathbf{f}, \mathbf{y}} \mathcal{W}(\mathbf{f}(\tilde{\theta}), \mathbf{y}(\tilde{\theta}), (\tilde{\theta}))$
$\mathbf{f}_{i=\phi}^*$	$\mathbf{f}_{-i}^* \cup (f_i = \phi)$

Table 2: Notations in Property Proofs

To summarize, the overall flow of the strategic RCPSP works as follows: (i) Agents bid for tasks of interest by reporting certain types. (ii) The central will then compute a task allocation together with a POS for project execution using the MILP model in Table 1. (iii) Agents perform allocated tasks following the POS. (iv) Payments are then made to individual agents (at different timings as in SCP and ICP).

4.4 Mechanism Properties

In this section, we outline the key properties of our SCP and ICP mechanisms: First, we show that SCP with payment rule in Eqn 2 is Bayes Nash incentive compatible, i.e., an agent always receives the highest utility by reporting its true type when all other agents are truth reporting. We then show that ICP is more powerful than SCP, as it allows for individual rationality and strategy proofness (or general incentive compatibility). Strategy proofness implies that being truthful is the dominant strategy irrespective of what other agents report. Finally, we show that both SCP and ICP ensure center's rationality and economic efficiency. The key notations in mechanism properties are summarized in Table 2.

Bayes Nash Incentive Compatibility of SCP

Consider an arbitrary agent i . Let the optimal allocation based on the true types $\theta = (\theta^i, \theta^{-i})$ denoted by $\mathbf{f}^*(\theta)$, or \mathbf{f}^* . From the design of SCP, \mathbf{f}^* optimizes the total welfare, given θ . Let \mathbf{f}_{-i}^* denote the optimal allocation based on θ^{-i} . From Eqns 2, 3, and 6, the utility of agent i by reporting θ^i is:

$$\begin{aligned} u^i(\theta) &= p_{SCP}^i(\theta) - C^i(\theta) \\ &= \mathcal{W}^{-i,i}(\mathbf{f}^*, \mathbf{y}^*, \theta) - \mathcal{W}^{-i}(\mathbf{f}_{-i}^*, \mathbf{y}_{-i}^*, \theta^{-i}) - C^i(\theta) \\ &= \mathcal{W}^{-i,i}(\mathbf{f}^*, \mathbf{y}^*, \theta) + V^i - C^i(\theta) - \mathcal{W}^{-i}(\mathbf{f}_{-i}^*, \mathbf{y}_{-i}^*, \theta^{-i}) \\ &= \mathcal{W}(\mathbf{f}^*, \mathbf{y}^*, \theta) - \mathcal{W}^{-i}(\mathbf{f}_{-i}^*, \mathbf{y}_{-i}^*, \theta^{-i}) \end{aligned} \quad (19)$$

where \mathcal{W} is the optimal welfare based on θ and \mathcal{W}^{-i} is the total welfare of all agents except i when i is not present.

We then consider the case when agent i reports $\hat{\theta}^i$, and other agents declare true types θ^{-i} . Let $\tilde{\mathbf{f}}^*$ denote the optimal allocation based on $\tilde{\theta} = (\hat{\theta}^i, \theta^{-i})$. Similarly in Eqn 19, the utility of agent i under allocation $\tilde{\mathbf{f}}^*$ can be represented as:

$$u^i(\tilde{\theta}) = \mathcal{W}(\tilde{\mathbf{f}}^*, \tilde{\mathbf{y}}^*, \theta) - \mathcal{W}^{-i}(\mathbf{f}_{-i}^*, \mathbf{y}_{-i}^*, \theta^{-i}) \quad (20)$$

where $\tilde{\mathbf{f}}^*$ is derived based on declarations $\tilde{\theta}$, while the welfare calculation involves true types θ . Thus, from Eqns 19 and 20,

$$u^i(\theta) - u^i(\tilde{\theta}) = \mathcal{W}(\mathbf{f}^*, \mathbf{y}^*, \theta) - \mathcal{W}(\tilde{\mathbf{f}}^*, \tilde{\mathbf{y}}^*, \theta) \quad (21)$$

Since $(\mathbf{f}^*, \mathbf{y}^*)$ optimizes the welfare based on θ , we then have $u^i(\theta) \geq u^i(\tilde{\theta})$. That is, given other agents are truthful, by truthfully declaring its type, agent i can get higher utility.

Individual Rationality of ICP

Individual Rationality requires that no agent would lose by participating with the true value, no matter what other agents declare. Consider agent i with its true type θ^i , and the reported types of other agents except i denoted by $\hat{\theta}^{-i}$. Let $\bar{\theta} = (\theta^i, \hat{\theta}^{-i})$ and the optimal allocation based on $\bar{\theta}$ denoted by $\bar{\mathbf{f}}^*(\bar{\theta})$, or $\bar{\mathbf{f}}^*$. From Eqns 4 - 6, the utility of agent i is:

$$\begin{aligned} u^i(\bar{\theta}) &= p_{ICP}^i(\bar{\theta}) - C^i(\bar{\theta}) \\ &= \mathcal{W}^{-i,i}(\bar{\mathbf{f}}^*, \bar{\mathbf{y}}^*, \bar{\theta}) - \mathcal{W}^{-i}(\mathbf{f}_{-i}^*, \mathbf{y}_{-i}^*, \hat{\theta}^{-i}) - C^i(\bar{\theta}) \\ &= \mathcal{W}^{-i|i}(\bar{\mathbf{f}}^*, \bar{\mathbf{y}}^*, \bar{\theta}) + V^i - C^i(\bar{\theta}) - \mathcal{W}^{-i}(\mathbf{f}_{-i}^*, \mathbf{y}_{-i}^*, \hat{\theta}^{-i}) \\ &= \mathcal{W}(\bar{\mathbf{f}}^*, \bar{\mathbf{y}}^*, \bar{\theta}) - \mathcal{W}^{-i}(\hat{\mathbf{f}}_{-i}^*, \hat{\mathbf{y}}_{-i}^*, \hat{\theta}^{-i}) \end{aligned} \quad (22)$$

where \mathcal{W}^{-i} is the total welfare of all agents except i when i is not present and the optimal allocation $\hat{\mathbf{f}}_{-i}^*$ is derived based on $\hat{\theta}^{-i}$. Note that in \mathcal{W}^{-i} , since i is not involved, the welfare remains unchanged if we append an empty allocation for i to $\hat{\mathbf{f}}_{-i}^*$, with $\hat{\mathbf{f}}_{i=\phi}^* = \hat{\mathbf{f}}_{-i}^* \cup (\hat{f}_i = \phi)$. That is,

$$u^i(\bar{\theta}) = \mathcal{W}(\bar{\mathbf{f}}^*, \bar{\mathbf{y}}^*, \bar{\theta}) - \mathcal{W}(\hat{\mathbf{f}}_{i=\phi}^*, \hat{\mathbf{y}}_{i=\phi}^*, \hat{\theta}^{-i}) \quad (23)$$

where the last term represents the total welfare of all agents (including i with zero contribution) based on allocation $\hat{\mathbf{f}}_{i=\phi}^*$ (derived from $\hat{\theta}^{-i}$). Given $\hat{\mathbf{f}}_{i=\phi}^*$, the welfare function remain unchanged irrespective of the type of agent i . Thus, we have,

$$u^i(\bar{\theta}) = \mathcal{W}(\bar{\mathbf{f}}^*, \bar{\mathbf{y}}^*, \bar{\theta}) - \mathcal{W}(\hat{\mathbf{f}}_{i=\phi}^*, \hat{\mathbf{y}}_{i=\phi}^*, \bar{\theta}) \quad (24)$$

Since $(\bar{\mathbf{f}}^*, \bar{\mathbf{y}}^*)$ optimizes \mathcal{W} given $\bar{\theta}$, we then have $u^i(\bar{\theta}) \geq 0$. That is, agent i 's utility, no matter what other agents report, if it truthfully declares its type, is always non-negative.

Strategy Proofness of ICP

Consider i reports $\hat{\theta}^i$ and other agents declare $\hat{\theta}^{-i}$. According to Eqn 4 and similar derivations from Eqns 22 to 24, the utility of i in ICP mechanism based on $\hat{\theta} = (\hat{\theta}^i, \hat{\theta}^{-i})$ is,

$$\bar{u}^i(\hat{\theta}) = \mathcal{W}(\hat{\mathbf{f}}^*, \hat{\mathbf{y}}^*, \hat{\theta}) - \mathcal{W}(\hat{\mathbf{f}}_{i=\phi}^*, \hat{\mathbf{y}}_{i=\phi}^*, \hat{\theta}) \quad (25)$$

And the utility based on $\bar{\theta} = (\theta^i, \hat{\theta}^{-i})$ can be represented as,

$$\bar{u}^i(\bar{\theta}) = \mathcal{W}(\bar{\mathbf{f}}^*, \bar{\mathbf{y}}^*, \bar{\theta}) - \mathcal{W}(\hat{\mathbf{f}}_{i=\phi}^*, \hat{\mathbf{y}}_{i=\phi}^*, \bar{\theta}) \quad (26)$$

Note that the last items in Eqns 25 and 26 are the same, given agent i has empty allocation. Thus,

$$\bar{u}^i(\bar{\theta}) - \bar{u}^i(\hat{\theta}) = \mathcal{W}(\bar{\mathbf{f}}^*, \bar{\mathbf{y}}^*, \bar{\theta}) - \mathcal{W}(\hat{\mathbf{f}}^*, \hat{\mathbf{y}}^*, \bar{\theta}) \quad (27)$$

Since $(\bar{\mathbf{f}}^*, \bar{\mathbf{y}}^*)$ is the optimal allocation based on $\bar{\theta}$, we then have $\bar{u}^i(\bar{\theta}) \geq \bar{u}^i(\hat{\theta})$. In other words, no matter what declaration other agents provide, by telling the truth, agent i can get the highest utility.

Center's Rationality and Economic Efficiency

A mechanism is said to satisfy center's rationality and economic efficiency if it guarantees that center has non-negative utility and the welfare is optimal if all participants are truthful, respectively. If all agents are truthfully declaring their types, there would be no difference between SCP and ICP in terms of center's utility and social welfares achieved. Thus, we only prove center's rationality and economic efficiency for the SCP mechanism and omit the proof for ICP.

From Eqn 7, the center's utility is the difference between the total value created by agents from attending the allocated tasks and the total payments made to agents. Consider all agents are truth reporting. From Eqn 2, the center's utility is,

$$\begin{aligned} u^\#(\theta) &= V - \sum_i p_{SCP}^i(\theta) \\ &= V - \sum_i \left(\mathcal{W}^{-i,i}(\mathbf{f}^*, \mathbf{y}^*, \theta) - \mathcal{W}^{-i}(\mathbf{f}_{-i}^*, \mathbf{y}_{-i}^*, \theta^{-i}) \right) \\ &= \sum_i \left(\mathcal{W}^{-i}(\mathbf{f}_{-i}^*, \mathbf{y}_{-i}^*, \theta^{-i}) - \mathcal{W}^{-i|i}(\mathbf{f}^*, \mathbf{y}^*, \theta) \right) \end{aligned} \quad (28)$$

where $\mathcal{W}^{-i|i}$ is the welfare of participants except i when i is present, \mathbf{f}^* and \mathbf{f}_{-i}^* are optimal allocations based on θ and θ^{-i} , respectively. Since \mathbf{f}_{-i}^* optimizes the welfare of all agents but i , we then have $u^\#(\theta) \geq 0$. That is, if all agents are truthful, the mechanism guarantees the center a non-negative utility.

Since \mathbf{f}^* optimizes the total welfare \mathcal{W} based on θ , economic efficiency can be directly achieved. That is, the welfare of all participants is optimal if all agents are truthfully declaring their types.

5 Conclusion

Existing research in RCPSP assumes that a central authority is equipped with all related data of the problem (e.g., processing times of tasks) and is asked to derive a scheduling solution that optimises a certain criteria (e.g. project makespan) with both temporal and resource constraints satisfied. But in the real world project scheduling, decisions might be taken by several individual economic units or agents, aiming at optimising their own objectives rather than the project performance. As such, agents may lie and declare values other than the truth for self interests. In this work, we first define a strategic RCPSP model and then propose mechanisms for the strategic RCPSP, that ensures agents are incentivised to reveal truth about their types and thereby executing the tasks on time to guarantee a successful project delivery.

References

- [Agnētis *et al.*, 2015] Alessandro Agnētis, Cyril Briand, Jean-Charles Billaut, and Premysl Sucha. Nash equilibria for the multi-agent project scheduling problem with controllable processing times. *Journal of Scheduling*, 18(1):15–27, 2015.
- [Archer and Tardos, 2007] Aaron Archer and Éva Tardos. Frugal path mechanisms. *ACM Transactions on Algorithms*, 3(1):3:1–3:22, February 2007.
- [Clarke, 1971] Edward Clarke. Multipart pricing of public goods. *Public Choice*, 11(1):17–33, 1971.
- [Dobzinski and Sundararajan, 2008] Shahar Dobzinski and Mukund Sundararajan. On characterizations of truthful mechanisms for combinatorial auctions and scheduling. In *Proceedings of the 9th ACM Conference on Electronic Commerce*, pages 38–47, 2008.
- [Estévez-Fernández, 2012] Arantza Estévez-Fernández. A game theoretical approach to sharing penalties and rewards in projects. *European Journal of Operational Research*, 216(3):647–657, 2012.
- [Fu *et al.*, 2012] Na Fu, Hoong Chuin Lau, Pradeep Varakantham, and Fei Xiao. Robust local search for solving rcpsp/max with durational uncertainty. *Journal of Artificial Intelligence Research*, 43:43–86, 2012.
- [Fu *et al.*, 2016] Na Fu, Pradeep Varakantham, and Hoong Chuin Lau. Robust partial order schedules for rcpsp/max with durational uncertainty. In *Proceedings of the 26th International Conference on Automated Planning and Scheduling*, pages 124–130, 2016.
- [Groves, 1973] Theodore Groves. Incentives in teams. *Econometrica*, 41(4):617–631, 1973.
- [Heidenreich *et al.*, 2007] Birgit Heidenreich, Rudolf Muller, and Marc Uetz. Games and mechanism design in machine scheduling - an introduction. *Production and Operations Management*, 16(4):437–454, 2007.
- [Kolisch and Sprecher, 1996] Rainer Kolisch and Arno Sprecher. Psplib - a project scheduling library. *European Journal of Operational Research*, 96:205–216, 1996.
- [Nisan and Ronen, 2001] Noam Nisan and Amir Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35(1-2):166–196, 2001.
- [Pinedo, 2008] Michael L. Pinedo. *Scheduling: Theory, Algorithms, and Systems*. Springer Publishing Company, Incorporated, 3rd edition, 2008.
- [Policella *et al.*, 2007] Nicola Policella, Amedeo Cesta, Angelo Oddi, and Stephen F. Smith. From precedence constraint posting to partial order schedules: A csp approach to robust scheduling. *AI Communications*, 20:163–180, August 2007.
- [Porter *et al.*, 2008] Ryan Porter, Amir Ronen, Yoav Shoham, and Moshe Tennenholtz. Fault tolerant mechanism design. *Artificial Intelligence*, 172(15):1783 – 1799, 2008.
- [Postlewaite and McLean, 2015] Andrew Postlewaite and Richard McLean. Implementation with interdependent valuations. *Theoretical Economics*, 10(3), 2015.
- [Schutt *et al.*, 2013] Andreas Schutt, Thibaut Feydy, Peter J. Stuckey, and Mark G. Wallace. Solving rcpsp/max by lazy clause generation. *Journal of Scheduling*, 16(3):273–289, 2013.
- [Skowron *et al.*, 2017] Piotr Skowron, Krzysztof Rzdca, and Anwitaman Datta. Cooperation and competition when bidding for complex projects: Centralized and decentralized perspectives. *IEEE Intelligent Systems*, 32(1):17–23, 2017.
- [Vanhoucke and Coelho, 2016] Mario Vanhoucke and José Coelho. An approach using sat solvers for the rcpsp with logical constraints. *European Journal of Operational Research*, 249(2):577–591, 2016.
- [Varakantham *et al.*, 2016] Pradeep Varakantham, Na Fu, and Hoong Chuin Lau. A proactive sampling approach to project scheduling under uncertainty. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 3195–3201, 2016.
- [Vickrey, 1961] William Vickrey. Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance*, 16(1):8–37, 1961.