Optimal Verification Strategies in Multi-Firm Projects[1]

Dr. Aditya U. Kulkarni (aditya88@vt.edu)

Grado Department of Industrial and Systems Engineering, Virginia Tech

Dr. Alejandro Salado (alejandrosalado@arizona.edu)

Department of Systems and Industrial Engineering, The University of Arizona

Dr. Christian Wernz (chris.wernz@gmail.com)

Department of Data Science, University of Virginia Health System

Abstract

Verification activities are intended to reduce the costs of system development by identifying design errors before deploying the system. However, subcontractors in multi-firm projects are motivated to implement locally cost-effective verification strategies over verification strategies that benefit the main contractor. Incentivizing verification activities is one mechanism by which the contractor can motivate subcontractors to implement verification strategies desirable to the contractor. Prior work on mathematical models of verification in systems engineering has neither explored optimal verification strategies nor incentives in multi-firm projects. In this paper, we present a modeling concept for determining optimal verification strategies in multi-firm projects. Our models are belief based, which means that contractors and subcontractors incorporate their at times limited knowledge about true verification state through a probabilistic assessment of possible states. We develop an initial two-level model, where one contractor directly works with multiple

[1] This article has been published: Kulkarni, AU, Salado, Alejandro, Wernz, Christian. Optimal verification strategies in multi-firm projects. *Systems Engineering*. 2022; 00 1– 17. https://doi.org/10.1002/sys.21615

subcontractors at the next lower level. This model is then extended to a general network model with multiple, multi-level contractor-subcontractor relationship. We derive solution algorithms that characterize the optimal verification strategies and incentives for each of the firms. Our work contributes to the systems engineering literature by laying the foundation for the study of incentives as a mechanism to align verification activities in multi-firm systems engineering projects.

Keywords: Verification, multi-firm, belief distributions, incentives

1 <u>Introduction</u>

Verification activities, such as design analyses, inspections and tests, help reduce development costs by identifying errors early in the development cycle. They seek confirmation that a system's behavior and attributes match those expected during its design and fabrication [1]. Verification activities are executed at several design integration levels and at different points throughout the system life cycle [2]. Implementing optimal project-wide verification strategies is thus key to maximizing the main contractor's rewards.

However, most firms do not follow such a comprehensive and structured approach to verification [3]. Consequently, verification activities consume a larger than necessary amount of resources during the system design process [4]. This problem is exacerbated when multiple firms are involved in the system design process, since planning and executing verification activities become increasingly complex [5].

In multi-firm projects, each firm is motivated by their individual interests. While the contractor seeks to improve the confidence in the correctness of the system design as a whole, each subcontracted firm's strategy is motivated by the maximization of their individual rewards, which may not align with that of the contractor [6]. To address this problem, the systems engineering research community has studied how incentives can overcome conflicting interests in multi-firm projects [7-11]. Prior work, however, has mainly focused on design activities, and not on verification.

In the supply chain literature, incentives have been widely studied for verification as an activity to improving product quality, e.g., [12-14]. However, this literature focuses mainly on the repeatable manufacturing of products with given designs. This is unlike most systems engineering projects where engineers develop systems with a high degree of novelty to meet mission requirements.

In this research article, we provide a foundational modeling approach for using incentives (monetary or otherwise) to align verification activities in multi-firm systems engineering projects. We contribute two belief-based models of verification: 1) a two-level model, where there are multiple subcontractors, each working on a critical component, for a main contractor, and 2) the network model, which models a general multi-firm systems engineering project with the firms organized in a hierarchy and each working on a critical component for the system. For each model, we present exact algorithms that can be used by a practitioner to determine optimal incentives and verification strategies. We illustrate our models with numerical examples that capture the benefits of incentivizing verification activities in multi-firm design projects, with one of the benefits being the maximization of the main contractor's expected reward. This paper extends our prior work on the two-firm, contractor-subcontractor model [15], which provided an initial concept on how the contractor can benefit from offering incentives to the subcontractor for verification activities, both in terms of expected reward and an improved confidence in the correctness of the system design.

As such, the work is restricted to the design of verification strategies and does not address the design of validation strategies.

The remainder of this paper is organized as follows. In section 2, we provide a brief overview of the literature on verification activities and our motivation for using belief distributions. In section 3, we provide a summary of our prior two-firm model [15], which forms the basic building block of the two-level and network models presented in this paper. We then develop the two-level model in section 4. Here, we provide an exact algorithm to compute optimal incentives for all subcontractors and illustrate the concepts with a numerical example. We extend the two-level model to the network model in section 5, where we provide an exact algorithm to compute the optimal incentives in the network model and illustrate the same with a numerical example. We conclude by summarizing key insights in section 6.

2 Background

The uncertainty about processes observed in nature are often due to variations in the underlying natural process. Such uncertainty is aleatory in nature, where a frequentist approach can be used to suitably represent the natural process as a stochastic process. Aleatory uncertainty is fundamentally different from the uncertainty observed in systems engineering projects, which is primarily epistemic in nature [16]. That is, the primary uncertainty in systems engineering projects is due to the lack of knowledge about the true state of the system design. Indeed, multiple works [17-19] have identified that designers maintain subjective beliefs about the true state of the system design during the design process. In this regard, verification activities mitigate the epistemic uncertainty on a system design by revealing more information about the true current state of the system design [20]. Yet, majority of the literature on verification activities adopts the aleatory uncertainty approach [3, 5, 21-37]. The reader is referred to [38, 39] for a discussion on these works and the drawbacks of adopting the aleatory uncertainty approach to model systems engineering projects.

Recent works on verification in systems engineering have begun to explore the benefits of modeling the epistemic uncertainty in systems engineering projects with belief distributions [38-47]. In these works, belief distributions are used to model the subjective confidence of designers in the true state of the system design. By leveraging Bayesian, Markovian, and machine learning frameworks in addition to belief distributions, these works have tried to uncover the scientific foundations of verification activities in systems engineering. A variety of fundamental research questions have been explored by these works, which include the fundamental nature of verification activities [42-44], eliciting beliefs from designers and using Bayesian inference populate the tradespace of verification strategies [40], and capturing the information dependencies between verification activities over the system's lifecycle with Bayesian networks [45].

In addition to exploring fundamental research questions on verification activities, a few of these belief-based approaches to verification have exploited algorithmic approaches to determine optimal verification strategies in systems engineering projects [38, 39, 47-49]. These include the use of reinforcement learning [47] and dynamic programming [38, 39] to explore the space of verification strategies in order to determine the optimal verifications strategy for a single firm. Our work contributes to this area of research by developing two initial concept models of verification in multi-firm projects using belief distributions and developing exact algorithms to determine optimal verification for the same.

Though it is tempting to assume complete cooperation in multi-firm projects, prior work has acknowledged that appropriate incentives are necessary to overcome self-interest in multi-firm projects [7-11]. However, to the best of our knowledge, other than in our recent work [15, 50-52], incentivizing verification activities in multi-firm projects has not been explored in systems engineering literature. There is, however, a significant number of works dealing with a similar problem in quality control literature for supply chains [12, 13, 53-55]. In this literature, the focus is mainly on those scenarios where products are mass produced, or a single order consists of multiple products of the same type. Furthermore, such works assume that verification activities are not contracted upon, and instead, the contracts only specify product quality level the supplier must meet [56].

As we observed in our prior work [15], verification in systems engineering projects is fundamentally different from verification in supply chains, since systems engineering projects often involve novel designs that are often complex and costly, and which require the participation of engineers from multiple disciplines. Furthermore, in systems engineering projects, the contractor may only discover an erroneous component design when the entire system design is verified (e.g., discovering errors in embedded systems through hardware-in-loop simulations). For this reason, we build our two-level and network models of verification by using our belief-based two-firm model presented in [15] as the basic building block for the two-level and network models, which builds upon multiscale decision theory (MSDT) [57].

3 Two-firm model

In this section, we present the two-firm model of incentives for verification [15]. Note that incentives are not restricted to monetary ones but can also be of non-monetary nature. The two-firm model forms the basic building block of the two-level and network models. For the sake of brevity, we restrict our discussion to the description and main results of the two-firm model. For a detailed discussion on the two-firm model, including parameter analysis, the reader is referred to [15].

3.1 Model environment and scope

The two-firm model consists of a contractor and subcontractor. The contractor is responsible for designing a system for the customer, and the contractor delegates the design of a system component to a subcontractor. The design process for a system is modeled as a series of development phases [38, 39]. In each development phase, a firm (contractor or subcontractor) is assumed to first carry out some design activities and can choose to carry out verification activities once the design activities have concluded. Examples of design activities include modeling, tradespace studies, mock-ups, prototypes, and fabrication of final components, while the examples of verification activities include testing, inspection, and analysis. Only verification activities are considered, not validation ones. Similar to [15], we restrict the model scope to a single development phase, which we will refer to as the phase of interest. This restriction is motivated by our work being an initial concept and to ensure mathematical tractability when the two-firm model is scaled to two-level and network models.

The two-firm model developed in [15] makes a significant restrictive assumption about the phase of interest: *the subcontractor's component design is integrated into the contractor's system design*. This integration of contractor and subcontractor designs could be an integration of models, simulations, prototypes, or the final fabricated components themselves. The purpose of assuming that the designs of the two firms will be integrated in the phase of interest is to capture the value added by the subcontractor's component design on the contractor's system design. This value then implicitly determines the potential increase in the contractor's expected reward if the subcontractor is incentivized to verify the component design, given that the subcontractor was initially inclined not to verify the component design.

In the two-firm model, verification of a firm's design is optional. It is usually the case that subcontractors verify their design prior to delivering it to the contractor. Hence, our assumption may appear to be unrealistic since a subcontractor may be contractually obligated to verify its component design prior to its integration into the system design. However, it is possible that due to a lack of understanding of the component design, the contractually required verification activities may be inadequate [46]. Thus, the choice to verify in two-firm model can also represent additional verification activities the contractor may require the subcontractor to execute to improve confidence in the component design. At the same time, the contractor may also decide to conduct a verification activity on its own that the subcontractor had already performed at its level. The same holds for the contractor with the customer requiring the contractor to perform certain verification activities. Since these additional verification activities are not contractually required, the subcontractor may be unwilling to execute them without incentives from the contractor, which in turn is in line with the motivation for the two-firm model. We assume that the incentives are quantifiable. This quantification could take the form of a subjective value for the loss of goodwill when a faulty design is delivered, an objective penalty levied by the customer on the firm for a faulty design, or something else, as long as it is quantifiable. The field of decision analysis provides means to quantify seemingly non-quantifiable variables [58].

We assume that if a firm decides to verify its design in the phase of interest, then all design errors will be discovered and rectified, where an error in design is a deviation from requirements. Though verification activities, in general, do not reveal all possible errors in an artifact design, this assumption was adopted for mathematical tractability [15]. In this paper, we adopt this assumption for the same reason.

For the subcontractor, verifying the component design implies that the subcontractor will discover all errors in the component design. Whereas, for the contractor, since the component design is integrated into the system design, verifying the system design implies that the contractor will discover errors in the component design as well. This is under the understanding that system integration and verification embed component validation; so the contractor may identify component errors that affect the ability of the system to fulfill its requirements. The implication that the contractor will discover errors in the component design, though restrictive, sets up the motivation for the contractor to incentivize the subcontractor to verify the component design, and potentially avoid costly rework activities on the system level.

The state of each firm's design is broadly classified as either ideal, or non-ideal. A design is said to be in the ideal state if it meets all its requirements and is said to be in the non-ideal state otherwise. Each state of a firm's design is associated with a state-based reward. Furthermore, each firm's verification activity is associated with a fixed set-up cost for verification and a potential expected cost of design rework if verification reveals an error in firm's design. That is, if a firm chooses to execute the verification of its design in the phase of interest, then it will certainly incur the set-up cost and will incur the expected cost of rework if verification reveals an error in its design.

Since prior to verification, neither firm can know the true state of its design (ideal or non-ideal), we use belief distributions [15, 38, 39, 44] to model a firm's knowledge in the state of its design. Then, the goal of each firm is to select the action that maximizes its expected rewards based on its belief in the ideal state of its design. We assume that each firm's belief is transformed during the design activities in the phase of interest. The factor by which each firm's belief in the ideal state of its design is transformed is modeled as the probability of a firm committing an error during the design activities, where an error is a feature of the artifact design that deviates from the artifact's requirements.

3.2 *Model description and notation*

In line with the notation used in [15], henceforth, we refer to the contractor as SUP and the subcontractor as INF. The phase of interest is represented by a time horizon. Each firm's time horizon begins with a mandatory design period and ends with the optional verification period. A firm's design is in its ideal state, denoted by 1, if the design meets all its requirements, else, it is considered to be in the non-ideal state, denoted by 0. Here, SUP's state variable denotes the state of the overall system design and INF's state variable denotes the state of the component design.

The start of a firm's time horizon is denoted by t_x, a firm's decision epoch (where it chooses to verify its design or not) is denoted by $\hat{t}_x$, and the end of the firm's time horizon is denoted by $\tilde{t}_x$, where $x \in \{\text{SUP}, \text{INF}\}$. At the end of INF's time horizon, SUP integrates INF's component design into the system design and this design integration occurs prior to SUP's decision epoch. SUP's timescale is thus longer than INF's time scale.

Each firm receives state-based rewards at the end of its time horizon. SUP covers the state-based rewards of INF and has the option to incentivize INF to verify its component design when INF is not inclined to verify its component design. A firm receives a reward of g_x if its design is in the ideal state and a reward of l_x if its design is in the non-ideal state at the end of its time horizon. Since the design phase is mandatory, we assume that the rewards of both firms are normalized with respect to their design costs. Verification costs, however, are dependent on a firm's decision to verify its design or not. If a firm chooses to verify its design, denoted by v_x, then it incurs a fixed setup cost of c_x, and an expected repair cost of r_x if any errors are present in the design. No costs are incurred by a firm if it chooses not to verify its design, and this action denoted by $-v_x$.

The design and verification activities of the INF firm influence the overall system design and verification activities executed by SUP. Similar to [15], we use the MSDT modeling approach to mathematically model the value added by INF's activities on SUP's activities as follows. Let $S_x, \hat{S}_x$, and $\tilde{S}_x$ be the state variables that denote the state of a firm's design at time $t_x, \hat{t}_x$, and $\tilde{t}_x$, respectively, with $S_x, \hat{S}_x, \tilde{S}_x \in \{0,1\}$. Here, $S_{\text{SUP}}, \hat{S}_{\text{SUP}}$, and $\tilde{S}_{\text{SUP}}$ represent the state of the overall system design, whereas, $S_{\text{INF}}, \hat{S}_{\text{INF}}$, and $\tilde{S}_{\text{INF}}$ represent the state of INF's component design. Let ε_{SUP} denote the probability of SUP making a design error when it chooses not to delegate any design tasks to INF. The following equation models the influence of INF's activities on SUP's final system design

$$p(\hat{S}_{\text{SUP}} = 0 \mid \tilde{S}_{\text{INF}}, S_{\text{SUP}} = 1, \text{delegation}) = p(\hat{S}_{\text{SUP}} = 0 \mid S_{\text{SUP}} = 1, \text{no-delegation}) + f(\hat{S}_{\text{SUP}}, \tilde{S}_{\text{INF}})$$

$$\Rightarrow p(\hat{S}_{\text{SUP}} = 0 \mid \tilde{S}_{\text{INF}}, S_{\text{SUP}} = 1, \text{delegation}) = \varepsilon_{\text{SUP}} + f(\hat{S}_{\text{SUP}}, \tilde{S}_{\text{INF}}), \quad \text{where} \tag{1}$$

$$f(\hat{S}_{\text{SUP}}, \tilde{S}_{\text{INF}}) = \begin{cases} -\theta & \text{if } \hat{S}_{\text{SUP}} = 0 \text{ and } \tilde{S}_{\text{INF}} = 1 \\ 1 - \varepsilon_{\text{SUP}} & \text{if } \hat{S}_{\text{SUP}} = 0 \text{ and } \tilde{S}_{\text{INF}} = 0 \end{cases} \tag{2}$$

is the influence function. Note, that $f(0,0) = 1 - \varepsilon_{\text{SUP}}$ implies that INF works on a critical component for SUP. That is, if INF's component has a design error, then SUP's system design will certainly not meet one or more of its requirements.

Equations (1) and (2) essentially define $p(\hat{S}_{\mathrm{SUP}} = 0 \mid \cdot)$ as a linear model with $\varepsilon_{\mathrm{SUP}}$ and θ. Unlike prior works, such as Salado et al. [44, 45, 59], where $p(\hat{S}_{\mathrm{SUP}} = 0 \mid \cdot)$ would be defined using a stochastic matrix, we choose to define $p(\hat{S}_{\mathrm{SUP}} = 0 \mid \cdot)$ as a linear, as described in Kulkarni et al. [51]. Representing $p(\hat{S}_{\mathrm{SUP}} = 0 \mid \cdot)$ as a linear model enables us to scale the two-firm model to multiple firms and multiple hierarchical levels. Though a linearity assumption is restrictive, more general models of multi-firm projects are out of the scope of this paper.

The influence function $f(\cdot)$ quantifies the benefits of delegation. Since SUP works on fewer components when it delegates design tasks to INF, if INF's component has no design errors, then the probability of SUP's system design having an error is reduced from $\varepsilon_{\mathrm{SUP}}$ to $\varepsilon_{\mathrm{SUP}} - \theta$. Similarly, if INF's component has a design error, then the probability of SUP's system design having an error is increased from $\varepsilon_{\mathrm{SUP}}$ to $\varepsilon_{\mathrm{SUP}} + 1 - \varepsilon_{\mathrm{SUP}} = 1$. That is, if INF's component has a design error, then the contractor's system is certain to have a design error after integration.

To model a firm's confidence in the state of its design, we use belief distributions. A firm's belief in the ideal state of its design is denoted by β_x at the start of the time horizon, by $\hat{\beta}_x$ at the decision epoch, and by $\tilde{\beta}_x$ at the end of the time horizon. Design and verification activities transform a firm's belief in the ideal state of its design. INF's beliefs are governed only by INF's design and verification activities, whereas SUP's beliefs are governed by both SUP and INF's design and verification activities.

We denote the probability of INF making a design error, during its design phase, by $\varepsilon_{\mathrm{INF}}$. Thus, at INF's decision epoch, $\hat{\beta}_{\mathrm{INF}} = \beta_{\mathrm{INF}}(1 - \varepsilon_{\mathrm{INF}})$. If INF chooses to verify its design at $\hat{t}_{\mathrm{INF}}$, then INF finds and repairs all the errors in its design and $\tilde{\beta}_{\text{INF}} = 1$. Else, INF's belief in the ideal state of its design is unchanged after the design phase and $\tilde{\beta}_{\text{INF}} = \hat{\beta}_{\text{INF}}$.

Since INF's component is integrated into SUP's system design before $\hat{t}_{\text{SUP}}$, SUP's beliefs at $\hat{t}_{\text{SUP}}$ are affected by INF's beliefs at $\tilde{t}_{\text{INF}}$. Using equations (1) and (2), SUP's belief in the ideal state of the system design at SUP's decision epoch is defined by

$$\hat{\beta}_{\text{SUP}}(\tilde{\beta}_{\text{INF}}) = \beta_{\text{SUP}}\left(1 - \varepsilon_{\text{SUP}} + \theta\tilde{\beta}_{\text{INF}} - (1 - \varepsilon_{\text{SUP}})(1 - \tilde{\beta}_{\text{INF}})\right)$$

$$\Rightarrow \hat{\beta}_{\text{SUP}}(\tilde{\beta}_{\text{INF}}) = \beta_{\text{SUP}}(1 - \varepsilon_{\text{SUP}} + \theta_1)\tilde{\beta}_{\text{INF}}. \tag{3}$$

Finally, if SUP chooses to verify the system design at $\hat{t}_{\text{SUP}}$, then $\tilde{\beta}_{\text{SUP}} = 1$, else SUP's beliefs are unchanged after the completion of its design phase and $\tilde{\beta}_{\text{SUP}} = \hat{\beta}_{\text{SUP}}$. Figure 1 graphically depicts the two-firm model scenario.

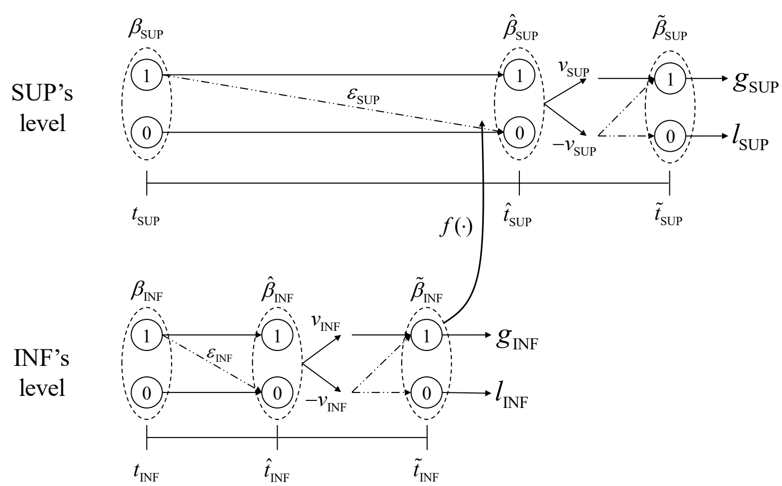


Figure 1: Graphical representation of the two-firm scenario

3.3 Optimal verification strategies[2]

[2] Note that this is a summary of the main results in [15].

The possible strategies for INF are to either verify or not verify its component design. SUP's strategy space consists of all feasible combinations of SUP's decision to either verify or not verify the system design and whether or not to incentivize INF to verify its component design.

3.3.1 Optimal verification strategy for INF without incentives

Denoting the rewards for INF by R_{INF}, the expected rewards for INF associated with its two possible choices, when SUP provides no additional incentives, are given by

$$\mathrm{E}(R_{\mathrm{INF}} \mid v_{\mathrm{INF}}, \beta_{\mathrm{INF}}) = g_{\mathrm{INF}} - c_{\mathrm{INF}} - r_{\mathrm{INF}}(1-\hat{\beta}_{\mathrm{INF}}), \text{ and} \tag{4}$$

$$\mathrm{E}(R_{\mathrm{INF}} \mid -v_{\mathrm{INF}}, \beta_{\mathrm{INF}}) = (g_{\mathrm{INF}} - l_{\mathrm{INF}})\hat{\beta}_{\mathrm{INF}} + l_{\mathrm{INF}}, \tag{5}$$

where $\hat{\beta}_{\mathrm{INF}} = \beta_{\mathrm{INF}}(1-\varepsilon_{\mathrm{INF}})$. Only if $\mathrm{E}(R_{\mathrm{INF}} \mid v_{\mathrm{INF}}, \beta_{\mathrm{INF}}) \geq \mathrm{E}(R_{\mathrm{INF}} \mid -v_{\mathrm{INF}}, \beta_{\mathrm{INF}})$, will INF verify its component design. This implies that INF will verify its component design when

$$\hat{\beta}_{\mathrm{INF}} \leq \left(1 - \frac{c_{\mathrm{INF}}}{g_{\mathrm{INF}} - l_{\mathrm{INF}} - r_{\mathrm{INF}}}\right) = \hat{\beta}^{*}_{\mathrm{INF}}. \tag{6}$$

The indifference threshold, denoted by β^{*}_{INF} in equation (6), is the belief threshold at which INF is indifferent between verifying and not verifying its design. We say that INF is of type confident if $\hat{\beta}_{\mathrm{INF}} \in (\beta^{*}_{\mathrm{INF}}, 1]$ and of type not-confident if $\hat{\beta}_{\mathrm{INF}} \in [0, \beta^{*}_{\mathrm{INF}}]$. Thus, when SUP offers no additional incentives, the optimal strategy for INF is to verify its design if it is of type not-confident at the end of the design phase.

3.3.2 Optimal incentive for a confident type INF

Since INF will verify its design without any additional incentives from SUP if it is of type not-confident, SUP need only consider incentivizing INF when INF is of type confident. For the two-firm model, we assume that any firm in the role of an INF will report its beliefs truthfully to its associated SUP. Though, this is a restrictive assumption, Kulkarni et al. [15] have shown that there exist incentive mechanisms by which SUP can offer the optimal incentive to a confident type INF, while discouraging a not-confident type INF from falsifying its belief in the ideal state of the component design and eliciting an incentive from SUP. With no incentives to lie about its beliefs, INF is assumed to report its beliefs truthfully to SUP.

To alter a confident type INF's strategy, SUP can affect the value of β^*_{INF} via incentives since it is a function of INF's rewards and costs. The optimal incentive is then that which changes INF's indifference threshold from β^*_{INF} to $\overline{\beta}^*_{\mathrm{INF}} = \hat{\beta}_{\mathrm{INF}}$, when $\hat{\beta}_{\mathrm{INF}} \in (\beta^*_{\mathrm{INF}}, 1]$. By shifting INF's threshold from β^*_{INF} to $\overline{\beta}^*_{\mathrm{INF}} = \hat{\beta}_{\mathrm{INF}}$, SUP effectively converts a confident type INF agent to a not-confident type INF agent and also offers the minimum incentive required to change INF's strategy from not verifying to verifying INF's design.

Let $z_{\mathrm{INF}} \in \{0,1\}$, with $z_{\mathrm{INF}} = 1$ denoting SUP incentivizes INF and $z_{\mathrm{INF}} = 0$ denoting SUP does not incentivize INF. Let $i_{\mathrm{INF}}(\hat{\beta}_{\mathrm{INF}})$ denote the optimal incentive amount that SUP must offer to INF when INF is of type confident. The expected reward for a confident type INF when SUP offers it incentives to verify its component design is defined by

$$\mathrm{E}(R_{\mathrm{INF}} \mid v_{\mathrm{INF}}, \beta_{\mathrm{INF}}, z_{\mathrm{INF}} = 1) = g_{\mathrm{INF}} - c_{\mathrm{INF}} - r_{\mathrm{INF}}(1 - \hat{\beta}_{\mathrm{INF}}) + i_{\mathrm{INF}}(\hat{\beta}_{\mathrm{INF}}). \tag{7}$$

Since the optimal incentive shifts the belief threshold from β^*_{INF} to $\overline{\beta}^*_{\mathrm{INF}} = \hat{\beta}_{\mathrm{INF}}$, we know that

$$\mathrm{E}(R_{\mathrm{INF}} \mid v_{\mathrm{INF}}, \beta_{\mathrm{INF}}, z_{\mathrm{INF}} = 1) = \mathrm{E}(R_{\mathrm{INF}} \mid -v_{\mathrm{INF}}, \beta_{\mathrm{INF}}, z_{\mathrm{INF}} = 0)$$

$$\Rightarrow i_{\mathrm{INF}}(\hat{\beta}_{\mathrm{INF}}) = c_{\mathrm{INF}} + \big(r_{\mathrm{INF}} - (g_{\mathrm{INF}} - l_{\mathrm{INF}})\big)(1 - \hat{\beta}_{\mathrm{INF}}). \tag{8}$$

3.3.3 Optimal strategy for SUP

Let $d^*_{\mathrm{INF}}(z_{\mathrm{INF}}, \hat{\beta}_{\mathrm{INF}})$ denote INF's optimal verification strategy, where $d^*_{\mathrm{INF}}(\cdot) = v_{\mathrm{INF}}$ if either $\hat{\beta}_{\mathrm{INF}} \in [0, \hat{\beta}^*_{\mathrm{INF}}]$ or $z_{\mathrm{INF}} = 1$, and $d^*_{\mathrm{INF}}(\cdot) = -v_{\mathrm{INF}}$ if $\hat{\beta}_{\mathrm{INF}} \in (\hat{\beta}^*_{\mathrm{INF}}, 1]$ and $z_{\mathrm{INF}} = 0$. Denoting the rewards for SUP by R_{SUP}, the SUP's expected rewards for its possible strategies are given by

$$\begin{aligned}\mathrm{E}(R_{\mathrm{SUP}} \mid \beta_{\mathrm{SUP}}, \hat{\beta}_{\mathrm{INF}}, -v_{\mathrm{SUP}}, z_{\mathrm{INF}}, d^*_{\mathrm{INF}}) &= (g_{\mathrm{SUP}} - l_{\mathrm{SUP}})\beta_{\mathrm{SUP}}(1 - \varepsilon_{\mathrm{SUP}} + \theta)\tilde{\beta}^*_{\mathrm{INF}}(d^*_{\mathrm{INF}}) + l_{\mathrm{SUP}} \\ &\quad - z_{\mathrm{INF}}\, i_{\mathrm{INF}}(\hat{\beta}_{\mathrm{INF}}) - (1 - z_{\mathrm{INF}})\big(\delta_{d^*_{\mathrm{INF}}, -v_{\mathrm{INF}}} \mathrm{E}(R_{\mathrm{INF}} \mid -v_{\mathrm{INF}}, \beta_{\mathrm{INF}}) + \delta_{d^*_{\mathrm{INF}}, v_{\mathrm{INF}}} g_{\mathrm{INF}}\big), \text{ and} \qquad (9)\end{aligned}$$

$$\begin{aligned}\mathrm{E}(R_{\mathrm{SUP}} \mid \beta_{\mathrm{SUP}}, \hat{\beta}_{\mathrm{INF}}, v_{\mathrm{SUP}}, z_{\mathrm{INF}}, d^*_{\mathrm{INF}}) &= g_{\mathrm{SUP}} - c_{\mathrm{SUP}} - r_{\mathrm{SUP}}\big(1 - \beta_{\mathrm{SUP}}(1 - \varepsilon_{\mathrm{SUP}} + \theta)\tilde{\beta}^*_{\mathrm{INF}}(d^*_{\mathrm{INF}})\big) \\ &\quad - z_{\mathrm{INF}}\, i_{\mathrm{INF}}(\hat{\beta}_{\mathrm{INF}}) - (1 - z_{\mathrm{INF}})\big(\delta_{d^*_{\mathrm{INF}}, -v_{\mathrm{INF}}} \mathrm{E}(R_{\mathrm{INF}} \mid -v_{\mathrm{INF}}, \beta_{\mathrm{INF}}) + \delta_{d^*_{\mathrm{INF}}, v_{\mathrm{INF}}} g_{\mathrm{INF}}\big), \qquad (10)\end{aligned}$$

where $\tilde{\beta}^*_{\mathrm{INF}}(d^*_{\mathrm{INF}}) = 1$ if $d^*_{\mathrm{INF}}(\cdot) = v_{\mathrm{INF}}$ and $\tilde{\beta}^*_{\mathrm{INF}}(d^*_{\mathrm{INF}}) = \hat{\beta}_{\mathrm{INF}}$ if $d^*_{\mathrm{INF}}(\cdot) = -v_{\mathrm{INF}}$, and δ is the indicator variable defined by $\delta_{a,b} = 1$ when $a = b$ and $\delta_{a,b} = 0$ when $a \neq b$.

From equations (9) and (10) we know that SUP will verify its design only if

$$\beta_{\mathrm{SUP}}(1 - \varepsilon_{\mathrm{SUP}} + \theta)\tilde{\beta}^*_{\mathrm{INF}}(d^*_{\mathrm{INF}}) = \hat{\beta}_{\mathrm{SUP}}(\beta_{\mathrm{SUP}}, \tilde{\beta}^*_{\mathrm{INF}}) \leq \Big(1 - \frac{c_{\mathrm{INF}}}{g_{\mathrm{INF}} - l_{\mathrm{INF}} - r_{\mathrm{INF}}}\Big) = \hat{\beta}^*_{\mathrm{SUP}}, \qquad (11)$$

where $\hat{\beta}_{\mathrm{SUP}}(\beta_{\mathrm{SUP}}, \tilde{\beta}^*_{\mathrm{INF}})$ denotes SUP's belief in the ideal state of the system design at $\hat{t}_{\mathrm{SUP}}$ given INF's final belief in the ideal state of the component design is $\tilde{\beta}^*_{\mathrm{INF}}$ after considering SUP's decision to incentivize INF or not. We see that SUP's indifference threshold $\hat{\beta}^*_{\mathrm{SUP}}$ is independent of INF's rewards and incentives since SUP offers INF the same rewards and incentives irrespective of SUP's final strategy.

We say that SUP is of type not-confident, and will thus verify the system design, if $\hat{\beta}_{\text{SUP}}(\tilde{\beta}^*_{\text{INF}}) \in [0, \hat{\beta}^*_{\text{SUP}}]$ and that SUP is of type confident, and will thus not verify the system design if $\hat{\beta}_{\text{SUP}}(\tilde{\beta}^*_{\text{INF}}) \in (\hat{\beta}^*_{\text{SUP}}, 1]$. We denote SUP's optimal verification strategy for a given $\hat{\beta}_{\text{SUP}}$ by $d^*_{\text{SUP}}(\hat{\beta}_{\text{SUP}})$. Since the optimal verification strategy of SUP, $d^*_{\text{SUP}}(\cdot)$, is a function of SUP's incentive strategy, SUP's strategy space can be reduced to its incentive strategy alone. SUP's optimal incentive strategy z^*_{INF} is then defined by the equation

$$z^*_{\text{INF}} = \underset{z_{\text{INF}} \in \{0,1\}}{\arg\max}\, \text{E}(R_{\text{SUP}} \mid \hat{\beta}_{\text{INF}}, z_{\text{INF}}, d^*_{\text{INF}}, \hat{\beta}_{\text{SUP}}, d^*_{\text{SUP}}). \tag{12}$$

4 <u>Two-level model</u>

The model scope and assumptions for the two-level model are similar to those of the two-firm model with one addition: there are now n subcontractors working for the contractor. Each subcontractor works on a unique component design.

4.1 <u>Model description and notation</u>

We continue to refer to the main contractor as SUP, but we will refer to a generic INF firm as INF firm x, where $x \in \{1, \ldots, n\}$. Similar to the two-firm model, SUP covers the state-based rewards of INF firm x and SUP has the choice to offer incentives for verification to an INF firm x that is not inclined to verify its design.

The structure of the time horizon for all firms is the same as the two-firm model: a mandatory design phase followed by an optional verification phase. The component design of each INF firm is completed and integrated into the system design before SUP's decision epoch, $\hat{t}_{\text{SUP}}$. The state of a firm's design is again broadly classified as either ideal or non-ideal. In addition, we assume that the state of a given INF firm's design is independent of the state of any other INF firm's design. That is, each INF's component design is decoupled from the designs of the components of other INFs.

We will use the same notation as the two-firm model for SUP's rewards, costs, probability of design error, time horizon epochs, and beliefs. For the INF firms, however, we will use the subscript x to denote INF firm x's rewards, costs, probability of design error, time horizon epochs, and beliefs instead of the subscript INF used in the two-firm model.

4.2 Optimal verification strategies for all firms

From the results of the two-firm model, we know that for INF firm x in the two-level model there is a belief threshold, denoted by β_x^*, that determines INF firm x's optimal verification strategy without incentives given $\hat{\beta}_x$. For INF firm x, $\hat{\beta}_x = \beta_x(1-\varepsilon_x)$, and INF firm x will verify its design if

$$\hat{\beta}_x \le \left(1 - \frac{c_x}{g_x - l_x - r_x}\right) = \hat{\beta}_x^*. \tag{13}$$

We say that INF firm x is of type not-confident, and will thus verify its design without additional incentives, if $\hat{\beta}_x \in [0, \hat{\beta}_x^*]$, and that INF firm x is of type confident, and will thus not verify its design without additional incentives if $\hat{\beta}_x \in (\hat{\beta}_x^*, 1]$.

From the two-firm model, we know that SUP need only offer $i_x(\hat{\beta}_x)$ to a confident type INF agent in order to motivate it to verify its design. To define optimal INF firm strategies with incentives, let $z_x \in \{0,1\}$, with $z_x = 1$ implying that SUP incentivizes INF firm x and $z_x = 0$ implying that SUP does not incentivize INF firm x. Let $d_x^*(z_x, \hat{\beta}_x)$ denote the optimal strategy of

INF firm *x,* where $d_x^*(\cdot) = v_{\text{INF}}$ if $z_x = 1$ or $\hat{\beta}_x \in [0, \hat{\beta}_x^*]$ and $d_x^*(\cdot) = -v_{\text{INF}}$ if $\hat{\beta}_x \in (\hat{\beta}_x^*, 1]$ and $z_x = 0.$ In addition, we denote INF firm *x*'s final belief resulting from its optimal strategy with incentives by $\tilde{\beta}_x^*(d_x^*),$ where $\tilde{\beta}_x^*(d_x^*) = 1$ if $d_x^*(\cdot) = v_x$ and $\tilde{\beta}_x^*(d_x^*) = \hat{\beta}_x$ if $d_x^*(\cdot) = -v_x.$

Similar to the INF firms, SUP's verification strategy is governed by a belief threshold, denoted by $\hat{\beta}_{\text{SUP}}^*$, that is a function of SUP's rewards and costs, but independent of the rewards and incentives SUP offers to INF firms. Given $\hat{\beta}_{\text{SUP}}^*$, SUP's verification strategy is completely characterized by its belief in the ideal state of the system design at $\hat{t}_{\text{SUP}}$, or $\hat{\beta}_{\text{SUP}}$, which in turn is a function of the final INF firm beliefs in the ideal states of their respective designs.

To determine $\hat{\beta}_{\text{SUP}}$, we first define the influence function for the two-level model. Let ε_{SUP} denote the probability of SUP making a design error when it does not delegate any design tasks to an INF firm, and let $\theta_{j_1,\ldots,j_n}$ denote the value of the influence function when $\tilde{S}_1 = j_1, \ldots, \tilde{S}_n = j_n.$ Since all the INF firms work on critical designs we know

$$p(\hat{S}_{\text{SUP}} = 0 \mid \exists w \in \{1,\ldots,n\} \text{ such that } \tilde{S}_w = 0) = 1 \Rightarrow \theta_{j_1,\ldots,j_w=1,\ldots,j_n} = 1 - \varepsilon_{\text{SUP}}. \qquad (14)$$

Equation (14) implies that the influence exerted by the activities of the INF firms on SUP's activities is the same when one or more INF firms have an error in design since an error in one INF firm's design implies an error in the overall system design. To complete the definition of the influence function, we define the influence of the activities of the INF firms on SUP's activities when all INF firms design their components without any error as

$$p(\hat{S}_{\text{SUP}} = 0 \mid \tilde{S}_1 = \ldots = \tilde{S}_n = 1) = \varepsilon_{\text{SUP}} - \theta_{1,\ldots,1}. \qquad (15)$$

We will denote $\theta_{1,\ldots,1}$ by θ. The influence function for the two-level model is then defined as

$$f(\cdot) = \begin{cases} -\theta & \text{if } \tilde{S}_1 = \ldots = \tilde{S}_n = 1 \\ 1-\varepsilon_{\text{SUP}} & \text{otherwise} \end{cases}.$$

Given the vector of final INF firm beliefs $\tilde{\beta}_{\text{INF}} = (\tilde{\beta}_1, \ldots, \tilde{\beta}_n)$, from the definition of the influence function, we know that SUP's belief in the ideal state of the system design at its decision epoch is given defined by

$$\hat{\beta}_{\text{SUP}}(\beta_{\text{SUP}}, \tilde{\beta}_{\text{INF}}) = \beta_{\text{SUP}}(1-\varepsilon_{\text{SUP}}+\theta)\prod_{x=1}^{n}\tilde{\beta}_x. \tag{16}$$

Thus, we know that SUP will verify its design in the two-level scenario only if

$$\hat{\beta}_{\text{SUP}}(\beta_{\text{SUP}}, \tilde{\beta}_{\text{INF}}) \le \left(1 - \frac{c_{\text{INF}}}{g_{\text{INF}} - l_{\text{INF}} - r_{\text{INF}}}\right) = \hat{\beta}^*_{\text{SUP}}. \tag{17}$$

We say that SUP is of type confident if $\hat{\beta}_{\text{SUP}}(\beta_{\text{SUP}}, \tilde{\beta}_{\text{INF}}) \in [0, \hat{\beta}^*_{\text{SUP}}]$ and SUP is of type not-confident if $\hat{\beta}_{\text{SUP}}(\beta_{\text{SUP}}, \tilde{\beta}_{\text{INF}}) \in (\hat{\beta}^*_{\text{SUP}}, 1]$. SUP's optimal verification strategy is to then verify the system design if it is of type not-confident at $\hat{t}_{\text{SUP}}$.

4.3 Optimal incentive strategy for SUP

SUP's optimal verification strategy is a function of INF firm beliefs, which in turn is a function of the incentives offered by SUP to the INF firms. Thus, SUP's strategy space can be reduced to the space of feasible incentives. Let $Z = (z_1, \ldots, z_n)$ denote SUP's incentive strategy. For a given vector of INF beliefs at their respective decision epochs $\hat{\beta}_{\text{INF}} = (\hat{\beta}_1, \ldots, \hat{\beta}_n)$, let the vector of optimal INF firm strategies be denoted by $D^*(Z, \hat{\beta}_{\text{INF}}) = \left(d_1^*(z_1, \hat{\beta}_1), \ldots, d_n(z_n, \hat{\beta}_n)\right)$. We denote the vector of final INF firm beliefs resulting from D^* by $\tilde{\beta}^*_{\text{INF}}(D^*) = \left(\tilde{\beta}^*_1(d^*_1),\ldots,\tilde{\beta}^*_n(d^*_n)\right)$. In addition, let $d_{\text{SUP}} \in \{v_{\text{SUP}}, -v_{\text{SUP}}\}$. SUP's expected rewards for strategy d_{SUP} is then defined by

$$\mathrm{E}(R_{\text{SUP}} \mid \beta_{\text{SUP}}, \hat{\beta}_{\text{INF}}, Z, D^*, \tilde{\beta}^*_{\text{INF}}, d_{\text{SUP}}) = \delta_{d_{\text{SUP}},-v_{\text{SUP}}}\left((g_{\text{SUP}} - l_{\text{SUP}})\beta_{\text{SUP}}(1-\varepsilon_{\text{SUP}}+\theta)\prod_{x=1}^{n}\tilde{\beta}^*_x + l_{\text{SUP}}\right)$$

$$+\delta_{d_{\text{SUP}},v_{\text{SUP}}}\left(g_{\text{SUP}} - c_{\text{SUP}} - r_{\text{SUP}}\left(1-\beta_{\text{SUP}}(1-\varepsilon_{\text{SUP}}+\theta)\prod_{x=1}^{n}\tilde{\beta}^*_x\right)\right)$$

$$-\sum_{x=1}^{n}\left(z_x\, i_x(\hat{\beta}_x) + (1-z_x)\left(\delta_{d^*_x,-v_x}\mathrm{E}(R_x \mid \beta_x, -v_x) + \delta_{d^*_x,v_x} g_x\right)\right). \quad (18)$$

SUP's belief in the ideal state of the system design at $\hat{t}_{\text{SUP}}$ resulting from $\tilde{\beta}^*_{\text{INF}}$ is denoted by $\hat{\beta}_{\text{SUP}}(\beta_{\text{SUP}}, \tilde{\beta}^*_{\text{INF}})$. We denote SUP's optimal verification strategy by $d^*_{\text{SUP}}(\hat{\beta}_{\text{SUP}})$, where $d^*_{\text{SUP}}(\cdot) = v_{\text{SUP}}$ if $\hat{\beta}_{\text{SUP}}(\cdot) \in [0, \hat{\beta}^*_{\text{SUP}}]$ and $d^*_{\text{SUP}}(\cdot) = -v_{\text{SUP}}$ if $\hat{\beta}_{\text{SUP}} \in (\hat{\beta}^*_{\text{SUP}}, 1]$. Then, SUP's optimal incentive strategy, denoted by Z^*, solves SUP's incentive problem $\mathbb{O}_T$ defined by

$$\max_{Z\in\{0,1\}^n} \mathrm{E}(R_{\text{SUP}} \mid \beta_{\text{SUP}}, \hat{\beta}_{\text{INF}}, Z, D^*, \tilde{\beta}^*_{\text{INF}}, d^*_{\text{SUP}}).$$

From equation (16) we know that $\mathbb{O}_{\text{T}}$ is a nonlinear 0-1 integer programming problem which is known to be an NP-hard problem [60].

4.4 Exact algorithm for SUP's incentive problem

We will exploit the problem structure to derive an exact algorithm for $\mathbb{O}_{\text{T}}$. Toward this end, when SUP does not incentivize any INF firm, let V_0 denote the set of INF firms that are of type not-confident and let $\overline{V_0}$ denote the set of INF firms that are of type confident. Let $I_c \subseteq \overline{V_0}$ denote the set of confident type INF firms currently incentivized by SUP and let $\bar{V}_c = \bar{V}_0 \setminus I_c$ denote the set of confident type INF firms not currently incentivized by SUP. Let $G_{c,k} \subseteq 2^{\bar{V}_c}$ denote the collection of all *k*-subsets of $\bar{V}_c$, where a *k*-subset is a subset consisting of *k* elements. Finally, let Z_A denote an incentive strategy vector for SUP such that $z_w = 1 \; \forall w \in A$ and $Z_A = \mathbf{0}$ if $A = \varnothing$.

From the definitions presented above, we know that the current set of INF firms that verify are $I_c \cup V_0$ and $V_0 \cup \bar{V}_0 = V_0 \cup I_c \cup \bar{V}_c = \{1, \ldots, n\}$. Given I_c, we say that it is profitable for SUP to incentivize INF firms in $A \subseteq \bar{V}_c$ if $\mathrm{E}(R_{\mathrm{SUP}} \mid \ldots, Z_{I_c \cup A}, \ldots, d^*_{\mathrm{SUP}}) \geq \mathrm{E}(R_{\mathrm{SUP}} \mid \ldots, Z_{I_c}, \ldots, d^*_{\mathrm{SUP}})$.

We begin by determining the best incentive strategy for SUP when it finds it profitable to incentivize a set of INF firms individually.

<u>Theorem 1</u> For a given d_{SUP}, if it is profitable for SUP to incentivize a set of INF firms $x_1, \ldots, x_m$ individually, where $1 \leq m \leq n$, then incentivizing all INF firms in the set $\{x_1, \ldots, x_m\}$ is a part of SUP's optimal incentive strategy. □

Proof of Theorem 1 is provided in Appendix A. Theorem 1 implies that each time SUP finds it profitable to incentivize a single INF firm x_j, the search space can be reduced to $\bar{V}_c \setminus \{x_j\}$ from $\bar{V}_c$, since incentivizing firm x_j will be a part of SUP's optimal incentive strategy. However, Theorem 1 only provides a sufficient condition for reducing search space and not a necessary one. Thus, it is possible for SUP that incentivizing all firms in $A \subseteq \bar{V}_c$, where $|A| \geq 2$, is part of SUP's optimal incentive strategy even if it is true that SUP does not find it profitable to incentivize each firm in A individually.

Consider the scenario where it is not profitable for SUP to incentivize any individual INF firm, and it is profitable for SUP to incentivize each set of INF firms $A_1,\ldots,A_m \subseteq 2^{\bar{V}_c} \setminus G_{c,1}$, where A_j is a collection of two or more INF firms for $j=1,\ldots,m.$ Here, the best strategy for SUP is to choose the set A^* such that $A^* = \arg\max_{A=\{A_1,\ldots,A_m\}} \mathrm{E}(R_{\mathrm{SUP}} \mid \ldots, Z_{I_c \cup A}, \ldots, d^*_{\mathrm{SUP}})$ for A^* defines the optimal set of firms to incentivize in $\bar{V}_c$, and incentivizing any set of INF firms other than A^* can potentially result in a sub-optimal incentive strategy. The implication of this is that if SUP finds no firm individually profitable in $\bar{V}_c$, then SUP has to search through $|2^{\bar{V}_c} \setminus G_{c,1}|$ combinations of INF firms to determine the optimal incentive strategy. Using Theorem 1 and the concepts presented above, we now define the exact algorithm to solve $\mathbb{O}_T$.

Exact algorithm for $\mathbb{O}_T$	
<u>*Input*</u>	
	$\bar{V}_0$
<u>*Initialize*</u>	
	$\bar{V}_c = \bar{V}_0, I_c = \varnothing, Z_{I_c}, \hat{\beta}_{\mathrm{INF}}, D^*, \tilde{\beta}^*_{\mathrm{INF}}, d^*_{\mathrm{SUP}}$
<u>*Execute*</u>	
1	Set $R_{\mathrm{SUP},c} = \mathrm{E}(R_{\mathrm{SUP}} \mid \beta_{\mathrm{SUP}}, \hat{\beta}_{\mathrm{INF}}, Z, D^*, \tilde{\beta}^*_{\mathrm{INF}}, d^*_{\mathrm{SUP}})$
2	Determine $\Omega_1 \subset G_{c,1}$ such that for $x \in \Omega_1$, $\mathrm{E}(R_{\mathrm{SUP}} \mid \beta_{\mathrm{SUP}}, \hat{\beta}_{\mathrm{INF}}, Z_{\{x\}}, D^*, \tilde{\beta}^*_{\mathrm{INF}}, d^*_{\mathrm{SUP}}) \geq R_{\mathrm{SUP},c}$
3	Set $I_c = \Omega_1, \bar{V}_c = \bar{V}_c \setminus \Omega_1$
4	If $\bar{V}_c = \varnothing$, set $I^*_c = I_c$ and go to <u>*Return*</u>
5	Using Z_{I_c}, update $\hat{\beta}_{\mathrm{INF}}, D^*, \tilde{\beta}^*_{\mathrm{INF}}, d^*_{\mathrm{SUP}}$
6	Set $R_{\mathrm{SUP},c} = \mathrm{E}(R_{\mathrm{SUP}} \mid \beta_{\mathrm{SUP}}, \hat{\beta}_{\mathrm{INF}}, Z, D^*, \tilde{\beta}^*_{\mathrm{INF}}, d^*_{\mathrm{SUP}})$
7	Solve $A^* = \arg\max_{A \in 2^{\bar{V}_c} \setminus G_{c,1}} \mathrm{E}(R_{\mathrm{SUP}} \mid \beta_{\mathrm{SUP}}, \hat{\beta}_{\mathrm{INF}}, Z_{I_c \cup A}, D^*, \tilde{\beta}^*_{\mathrm{INF}}, d^*_{\mathrm{SUP}})$
8	If $\mathrm{E}(R_{\mathrm{SUP}} \mid \beta_{\mathrm{SUP}}, \hat{\beta}_{\mathrm{INF}}, Z_{I_c \cup A^*}, D^*, \tilde{\beta}^*_{\mathrm{INF}}, d^*_{\mathrm{SUP}}) \geq R_{\mathrm{SUP},c}$, then set $I^*_c = I_c \cup A^*$.

Return I_c^*

4.5 Numerical example

There are 10 subcontractors, INF firms, working under the supervision of a single contractor, SUP. Each INF firm works on a unique and critical component design for SUP. For $x \in \{1,\ldots,10\}$, let q_x denote the probability of SUP making a design error if it chose to design the component delegated to INF firm x. Furthermore, let $q_{\text{SUP}} = 0.1$ be the probability of SUP making a design error in the components it does not delegate to any INF firm. From the values of q_x and q_{SUP}, we know

$$\varepsilon_{\text{SUP}} = 1-(1-q_{\text{SUP}})\prod_{x=1}^{n}(1-q_x) = 0.9523. \tag{19}$$

When SUP delegates the 10 components to the INFs, $\varepsilon_{\text{SUP}} = q_{\text{SUP}} = 0.1$. Thus, $\theta = 0.8523$.

The parameter values associated with the INF firms we consider for this example are presented in Table 1, and the parameter values associated with SUP that we consider for this example are presented in Table 2. The optimal verification strategies for all firms, with and without incentives, is presented graphically in Figure 2.

Table 1: INF firm parameters for two-level model

INF firm	g_x	l_x	c_x	r_x	ε_x	q_x	β_x	$\hat{\beta}_x^*$
1	400	200	50	50	0.1	0.28	1	0.667
2	400	200	50	60	0.1	0.29	1	0.643
3	400	200	50	10	0.1	0.26	1	0.737
4	400	200	80	80	0.2	0.27	0.1	0.333
5	400	200	50	60	0.2	0.27	0.1	0.643
6	400	200	130	60	0.1	0.23	1	0.071

7	400	200	120	70	0.1	0.26	1	0.077
8	400	200	130	50	0.3	0.21	1	0.133
9	800	200	400	170	0.1	0.27	1	0.07
10	800	200	500	90	0.1	0.20	1	0.012

Note: g_x : reward if in ideal state; l_x: reward if in non-ideal state; c_x : fixed setup costs of a verification activity; r_x : expected repair costs if error present; ε_x : probability of design error; q_x : probability of design error if SUP makes design of INF; β_x : belief in ideal state at the start of time horizon; $\hat{\beta}_x^*$: indifference threshold.

Table 2: SUP parameters for two-level model

g_{SUP}	l_{SUP}	c_{SUP}	r_{SUP}	ε_{SUP}	θ_1	β_{SUP}	$\hat{\beta}_{\text{SUP}}^*$
10,000	7,000	1,000	1,000	0.9523	0.8523	0.9	0.5

Note: g_{SUP} : reward in ideal state; l_{SUP} : reward if if non-ideal state; c_{SUP} : fixed setup cost to execute a verification activity; r_{SUP} : expected repair cost if design error is present; ε_{SUP} : probability of design error; θ_1 : value of influence function; β_{SUP} : belief in ideal state at start of the time horizon; $\hat{\beta}_{\text{SUP}}^*$: indifference threshold.

As shown in Figure 4, without incentives, only SUP and INF firms 4 and 5 verify their designs. INF firms 4 and 5 verify their design since their initial beliefs in the ideal state of their respective designs is lower than their respective belief thresholds at the start of the design phase, and thus they end up being not-confident type firms at the end of their respective design phase. Though SUP's initial belief in the ideal state of the system design is 1, which is greater than $\hat{\beta}_{\text{SUP}}^* = 0.5$, the combined influence of 8 INF firms not verifying their designs without incentives lowers SUP's belief in the ideal state of the system design, and prompts SUP to verify the system design when it doesn't offer any incentives to the INF firms.

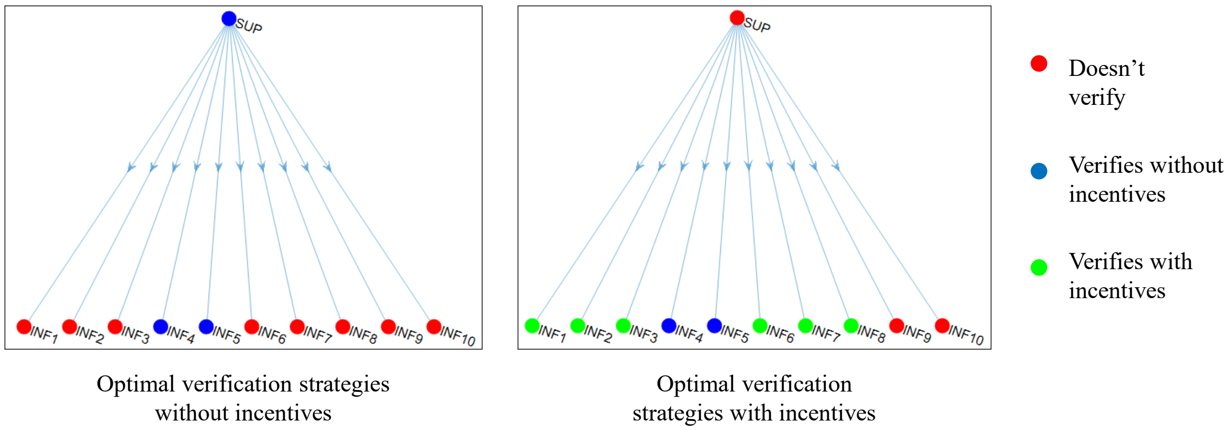


Figure 2: Two-level model solutions

When SUP is willing to incentivize the verification strategies of the INF firms, the optimal incentive strategy for SUP, as shown in figure 4, is to incentivize all INF firms except 9 and 10, with INF firms 4 and 5 verifying their designs without incentives. This results in all INF firms, except 9 and 10, verifying their design. The combined influence of INF firms $1,\ldots,8$ verifying their designs leads to SUP having sufficient confidence in the ideal state of the system design, and thus with incentives, SUP prefers not to verify the system design when it chooses to incentivize the INF firms. In addition, when SUP doesn't offer any incentives to the INF firms, SUP's expected reward is 4010. When SUP chooses to incentivize INF firms, its expected reward is 5169. Thus, by incentivizing the INF firms, SUP increases its expected rewards, after accounting for incentives, by approximately 22%, implying that incentivizing the INF firms to verify their designs maximizes SUP's rewards.

5 Network model

The two-level model consists of two hierarchical levels, one for the contractor and one for the subcontractors. The network model generalizes the two-level model to those scenarios where the subcontractors may hire subcontractors, and those subcontractors may hire additional subcontractors and so on. Due to multiple hierarchical levels, we will modify the notation used in the two-level model to ease the description of the network model.

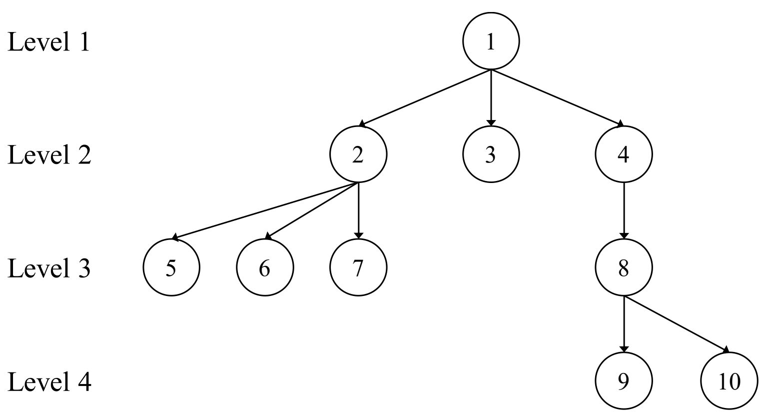
5.1 Model description

In the network model, there is one main contractor. Every other firm is a subcontractor of some other firm working in a supervisory role. There are n firms in the hierarchy and we refer to a generic firm as firm $x \in \{1,\ldots,n\}$. We use the subscript x to denote the model parameters associated with firm x. The time horizon for each firm in the hierarchy once again consists of a mandatory design phase followed by an optional verification phase, and the state of each firm's design is once again broadly classified as either ideal or non-ideal.

The number of hierarchical levels in the model is denoted by H and the hierarchical level of firm x is denoted by h_x, where $h_x \in \{1,\ldots,H\}$. The set of supervising firms, or firms that oversee the activities of at least one subordinate firm, is denoted by W and the set of subordinate firms, or firms that are supervised by another firm, is denoted by U. We refer to a generic supervising firm as firm $w \in W$ and a generic subordinate firm is referred to as firm $u \in U$.

We denote the set of immediate subordinates of a supervising firm w by T_w, and we assume that each subordinate firm has at most one immediate supervisor. In addition, we assume that the state of design of firm $x_1 \in T_w$ for any $w \in W$ is independent of the state of design of firm $x_2 \in T_w$ when $x_1 \neq x_2$. An immediate subordinate of a supervising firm w is the firm u such that $h_u = h_w + 1$ and $u \in T_w$, and the immediate supervisor for firm u is the firm w such that $h_w = h_u - 1$ and $u \in T_w$. It follows from the definitions above that no two supervising firms oversee same immediate subordinate, which in turn implies that $T_{w_1} \neq T_{w_2}$ for $w_1, w_2 \in W$ and $w_1 \neq w_2$, and the state of design of any firm $x_1 \in \{1,\ldots,n\}$ is independent of the state of design of any other firm $x_2 \in \{1,\ldots,n\}$, where $x_1 \neq x_2$, if no hierarchical path exists between x_1 and x_2.

In the network model, each firm in a supervising role will compensate the state-based rewards of its immediate subordinate and each supervising firm has the ability to incentivize a set of firms lower than itself in the hierarchy. The set of firms that a supervising firm w can potentially incentivize is referred to as firm w's set of control and is denoted by M_w. Since no two subordinate firms have the same immediate supervisor, it follows that $M_{w_1} \neq M_{w_2}$ for $w_1, w_2 \in W$ and $w_1 \neq w_2$. Figure 4 depicts the network model of a sample multi-firm project that consists of 10 firms including the main contractor.



$h_1 = 1, h_2 = h_3 = h_4 = 2,$
$h_5 = h_6 = h_7 = h_8 = 3$ and $h_9 = h_{10} = 4$.

$W = \{1,2,4,8\}$ and $U = \{2,\ldots,10\}$.

$T_1 = \{2,3,4\}, T_2 = \{5,6,7\}, T_4 = \{8\},$
$T_8 = \{9,10\}$.

Possible sets of control
$M_1 = \{2,3,4,5,6,7\}, M_2 = \{5,6,7\}$
$M_4 = \{8,9\}, M_8 = \{9,10\}$.

Figure 3: Example of a network representation of a multi-firm project

There is sequential delegation of critical component designs in the network model. The main contractor, firm 1, delegates the designs of a critical sub-system/component firms in T_1. The firms in T_1 then proceed to do the same with their subordinates and so on till the critical component designs are delegated to firms in $U \setminus W$, or the firms with no subordinates. Design integration occurs in the reverse order. The firms in $U \backslash W$ are the first to finish and their designs are integrated by their immediate supervisors before the decision epochs of these supervisors. This proceeds up the hierarchy till all firms in T_1 complete their designs and the main contractor integrates the designs of firms in T_1 into the system design before t_1. Thus, in the network model we have $\tilde{t}_x \le \hat{t}_w$ for all $x \in T_w$ and for all $w \in W$.

5.2 *Optimal verification strategies for all firms*

Without incentives, for firm x there is a belief threshold $\hat{\beta}_x^*$ such that firm x will verify its design if its belief in the ideal state of its design at $\hat{t}_x$, denoted by $\hat{\beta}_x$, is less than or equal to $\hat{\beta}_x^*$. We say that firm x is of type not-confident if $\hat{\beta}_x \in [0, \hat{\beta}_x^*]$ and it is of type confident if $\hat{\beta}_x \in (\hat{\beta}_x^*, 1]$. Thus, all not-confident type firms in the hierarchy will verify their design.

Firms in the set $U \backslash W$ have no subordinates and thus for any firm $x \in U \backslash W$, $\hat{\beta}_x = \beta_x(1-\varepsilon_x)$. The beliefs of a supervising firm w, however, is dependent on the final beliefs of firms in the set of its immediate subordinates T_w. Since all firms work on critical component designs, from the results of the two-level model we know that the influence function between firm w and the firms in T_w can be defined as

$$f_w(\cdot) = \begin{cases} -\theta_w & \text{if } \tilde{S}_x = 1 \; \forall x \in T_w \\ 1-\varepsilon_w & \text{otherwise} \end{cases}, \tag{20}$$

where ε_w is the probability of firm w making a design error when it does not delegate any design tasks to firms in T_w. Given the vector $\tilde{\beta}_{\text{INF},w} = (\tilde{\beta}_{x_1}, \ldots, \tilde{\beta}_{x_{|T_w|}})$ of final beliefs of firms immediately subordinate to firm w in the ideal state of their respective designs, the belief of a supervising firm w in the ideal state of its design at $\hat{t}_w$ is defined by

$$\hat{\beta}_w(\beta_w, \tilde{\beta}_{\mathrm{INF},w}) = \beta_w(1 - \varepsilon_w + \theta_w) \prod_{j \in \{1,\ldots,|T_w|\}} \tilde{\beta}_{x_j}. \tag{21}$$

The optimal verification strategy for each firm in the hierarchy is defined based on its type and whether or not it is incentivized to verify its design when it is of type confident. Let $z_u \in \{0,1\}$, with $z_u = 1$ denoting that firm u is incentivized to verify its design by some firm higher up in the hierarchy and $z_u = 0$ denoting that firm u is not incentivized to verify its design. For each subordinate firm u, let $\tilde{\beta}_u^*(d_u^*)$ denote firm u's final belief in the ideal state of its design that results from firm u's optimal verification strategy with incentives. A supervising firm w's belief at $\hat{t}_w$ is denoted by $\hat{\beta}_{w,*}$, where $\hat{\beta}_{w,*} = \hat{\beta}_w(\beta_w, \tilde{\beta}_{\mathrm{INF},w}^*)$. In addition, let $\hat{\beta}_{x,*} = \hat{\beta}_x$ for all $x \in U \setminus W$ since these firms have no subordinates. We then denote the optimal strategy of a subordinate firm u by $d_u^*(z_u, \hat{\beta}_{u,*})$, where $d_u^*(\cdot) = v_u$ if either $z_u = 1$ or $\hat{\beta}_{u,*} \in [0, \hat{\beta}_u^*]$ and $d_u^*(\cdot) = -v_u$ if $z_u = 0$ and $\hat{\beta}_{u,*} \in (\hat{\beta}_u^*, 1]$. Finally, the optimal strategy of firm 1 is denoted by $d_1^*(\hat{\beta}_{1,*})$ where $d_1^*(\cdot) = v_1$ if $\hat{\beta}_{1,*} \in [0, \hat{\beta}_1^*]$ and $d_1^* = -v_1$ otherwise.

5.3 <u>Optimal incentive strategy for a supervising firm</u>

In the network model if a firm $x_1 \in \{1,\ldots,n\}$ verifies its design, then it effectively verifies the design of firm $x_2 \in \{1,\ldots,n\}$, where $x_2 \neq x_1$ and x_2 is a firm lower than x_1 in the hierarchy such that there exists a path between x_1 and x_2 in the hierarchy. Thus, a supervising firm w need only consider incentivizing firms in the set $Y_w \subset M_w$ such that each firm $x \in Y_w$ is a confident type firm that is not currently incentivized and all firms on the hierarchical path between firms w and x are also confident type firms that are not currently incentivized. We refer to Y_w as the set of consideration for firm w.

We now formulate a supervising firm w's incentive problem. We only consider the case where $Y_w \neq \varnothing$ since $Y_w = \varnothing$ implies that all firms in the set of consideration of firm w will verify their design without any additional incentives from firm w. For ease of notation let $Y_w = (y_1, \ldots, y_{|Y_w|})$. We denote firm w's incentive strategy by $Z_w = (z_{y_1}, \ldots, z_{y_{|Y_w|}})$ and let the vector of subordinate firm beliefs at their decision epoch resulting from the incentive strategy Z_w be denoted by $\hat{\beta}_{\text{INF},w,*} = (\hat{\beta}_{y_1,*}, \ldots, \hat{\beta}_{y_{|Y_w|},*})$. The vector of optimal verification strategies of firms in the set of consideration for firm w that results from Z_w and $\hat{\beta}_{\text{INF},w,*}$ is denoted by $D_w^*(Z_w, \hat{\beta}_{\text{INF},w,*})$, where $D_w^*(Z_w, \hat{\beta}_{\text{INF},w,*}) = \left(d_{y_1}^*(z_{y_1}, \hat{\beta}_{y_1,*}), \ldots, d_{y_{|Y_w|}}^*(z_{y_{|Y_w|}}, \hat{\beta}_{y_{|Y_w|},*})\right)$. Finally, the vector of final beliefs of firms under w's depth of consideration in the ideal state of their design that results from D_w^* is denoted by $\tilde{\beta}_{\text{INF},w}^*(D_w^*) = \left(\tilde{\beta}_{y_1}^*(d_{y_1}^*), \ldots, \tilde{\beta}_{y_{|Y_w|}}^*(d_{y_{|Y_w|}}^*)\right)$.

From the results of the two-firm model, we know that firm w need only offer $i_j(\hat{\beta}_j)$ to a confident type firm j under firm w's depth of consideration in order to incentivize firm j to verify its design. Firm w's rewards associated with its verification strategy $d_w \in \{v_w, -v_w\}$ is defined by

$$\mathrm{E}(R_w \mid \beta_w, Z_w, \hat{\beta}_{\mathrm{INF},w,*}, D_w^*, \tilde{\beta}_{\mathrm{INF},w}^*, d_w) = \delta_{d_w,-v_w}\Big((g_w - l_w)\beta_w(1-\varepsilon_w+\theta_w)\prod_{x\in T_w\cap Y_w}\tilde{\beta}_x^* + l_w\Big)$$

$$+\delta_{d_w,-v_w}\Big(g_w - c_w - r_w\Big(1-\beta_w(1-\varepsilon_w+\theta_w)\tilde{\beta}_{w+1}^*\Big)\Big) - \sum_{j\in Y_w} z_j\; i_j\big(\hat{\beta}_{j,*}\big)$$

$$-\sum_{x\in T_w}\Big((1-z_x)\delta_{d_x^*,-v_x}\mathrm{E}(R_x \mid \hat{\beta}_{x,*}, -v_x) + \big((1-z_x)\delta_{d_x^*,v_x} + z_x\big)g_x\Big). \qquad (22)$$

Firm *w*'s optimal incentive strategy Z_w^* then solves the incentive problem $\mathbb{O}_{A,w}$ defined by

$$\max_{Z_w\in\{0,1\}^{|Y_w|}} \mathrm{E}(R_w \mid \beta_w, Z_w, \hat{\beta}_{\mathrm{INF},w,*}, D_w^*, \tilde{\beta}_{\mathrm{INF},w}^* d_w^*).$$

It follows from equations (21) and (22) that $\mathbb{O}_{A,w}$ is an NP-hard problem.

Each subordinate firm in the network model faces its decision epoch before its immediate supervising firm. This implies that for a supervising firm w, any firm $x \in T_w \cap W$ must solve its incentive problem $\mathbb{O}_{A,x}$ before firm w solves $\mathbb{O}_{A,w}$ so that firm w can determine Y_w, the set of consideration for firm w, appropriately. In addition, from the definition of the network model, we know that no two supervising firms oversee the same subordinate firm. This implies that if $x_1, x_2 \in T_w \cap W$, then it is irrelevant as to which firm, x_1 or x_2, solves their incentive problem first for the solution to $\mathbb{O}_{A,x_1}$ is independent of the solution to $\mathbb{O}_{A,x_2}$ due to $Y_{x_1} \neq Y_{x_2}$. Thus, the only requirement to determine the optimal incentive strategy for the entire hierarchy is for all firms $x \in T_w \cap W$ to solve their incentive problems before firm w solves its incentive problem.

5.4 Exact algorithm for a supervising firm's incentive problem

In the network model, a supervising firm w can potentially incentivize $|Y_w|$ firms below it in the hierarchy. Firm w must potentially evaluate $2^{|Y_w|}$ incentive strategies to determine the optimal incentive strategy. However, the structure of the network model can be exploited to narrow the search space to only those strategies that are potentially optimal and the search for the optimal incentive strategy can be optimized by utilizing the exact algorithm to solve $\mathbb{O}_T$ as follows.

Let $G_w \subset \{1,\ldots,n\} \setminus \{w\}$ denote those firms below a supervising firm w in the hierarchy such that if $x \in G_w$, then firm x's design is eventually integrated into firm w's design, and let $G_x = \varnothing$ for all $x \in U \setminus W$. The definition of G_w implies that for any firm $x \in G_w$, there exists a hierarchical path between firm x and firm w. We refer to G_x as the complete set of firm x's subordinates.

Any supervising firm in the network model can potentially minimize the incentives it offers to a firm $x \in Y_w$ by incentivizing the verification of one or more firms in $G_x \cap Y_w$. Toward this end, let $\Omega^*_{w,y} \subset G_y \cap Y_w$ be the set of firms such that if incentivizing firm y is a part of firm w's optimal incentive strategy, then incentivizing all firms in $\Omega^*_{w,y}$ is also part of firm w's optimal incentive strategy. Thus, if $y \in Y_w$, then firm w can ignore all incentive strategies with $z_y = 1$ and $z_x = 0$ for any $x \in \Omega^*_y$, where $x \neq y$.

For a supervising firm w, we say that a set of firms $A \subset G_w$ is a set of subordinate firms with complete coverage if $\hat{\beta}_w$ can be determined from the final beliefs of firms in A when no firm in G_w verifies its design. For the example presented in figure 4, for firm 1, given that no firm in G_1

verifies its design, the set of firms $\{2,3,4\}$ is a set of subordinate firms with complete coverage since

$$\hat{\beta}_1 = \beta_1(1-\varepsilon_1+\theta_1)\prod_{j=2}^{4}\tilde{\beta}_4. \tag{23}$$

The set of firms $\{5,6,7,3,4\}$ is also set of subordinate firms with complete coverage since $\tilde{\beta}_2$ in equation (23) can be replaced with $\tilde{\beta}_2 = \hat{\beta}_2 = \beta_2(1-\varepsilon_2+\theta_2)\prod_{j=5}^{7}\tilde{\beta}_j$. Thus, the remaining sets of subordinate firms with complete coverage for firm 1 are $\{2,3,8\}$, $\{5,6,7,3,8\}$, $\{2,3,9,10\}$, and $\{5,6,7,3,9,10\}$. Since no two supervising firm oversee the same subordinate and all firms work on unique critical component designs, it follows that the state of design for any firm in a set of subordinate firms with complete coverage is independent of the state of design of any other firm in the same set.

For a supervising firm w let $K_w \subset 2^{G_w}$ denote the collection of sets such that if $B \in K_w$, then B is a set of subordinate firms with complete coverage for firm w. For a supervising firm w and $B \in K_w$, let $\mathbb{O}_{T,w}(B)$ denote the solution to a two-level problem where w is SUP and the firms in set B are the INF firms with the firms in the set $B \cap Y_w$ not verifying their designs without additional incentives from the SUP firm w. The following proposition defines how $\mathbb{O}_{A,w}$ can be solved by utilizing the exact algorithm for $\mathbb{O}_T$.

<u>Proposition 1</u> The solution to the incentive problem for a supervising firm w in the network model, $\mathbb{O}_{A,w}$, is the solution to $\mathbb{O}_{T,w}(B)$ for some $B \in K_w$ when the search space for $\mathbb{O}_{T,w}(B)$ for all $B \in K_w$ is restricted to the incentives offered only to firms in $B \cap Y_w$ and $i_x(\hat{\beta}_{x,*}) = \sum_{j \in \Omega^*_{w,x}} i_j(\hat{\beta}_{j,*})$ for all $x \in B \cap Y_w$ □

The proof of the proposition 1 is constructed as follows. Since $\hat{\beta}_w$ can only be computed from the final beliefs of firms in $B \in K_w$, or the set of firms with complete coverage for firm w, we know that optimal incentive strategy for a supervising firm w can be determined by solving the two-level problem with w as SUP and the firms in $T_w = B_1 \in K_w$ as INF firms with the search space for the two-level problem restricted to the firms in $B_1 \cap Y_w$. If this is not true, then it must be true that firm w's optimal incentive strategy requires firm w to not incentivize some firm $x \in T_w \cap Y_w$. To maintain complete coverage, we must replace firm x with the firms in $T_x \cap Y_w$. But this leads to another set of firms $B_2 \in K_w$. Proceeding in a similar fashion we see that each time the set of subordinate firms in the two-level problem is equal to some set $B \in K_w$, which in turn proves proposition 3.

Proposition 1 implies that the exact algorithm to solve $\mathbb{O}_{A,w}$ is essentially solving $\mathbb{O}_{T,w}(B)$ for all $B \in K_w$ provided that for each $\mathbb{O}_{T,w}(B)$, the value of z_x for any firm $x \notin B \cap Y_w$ be unchanged and the cost of incentivizing firm $x \in B$ be set to $\sum_{j \in \Omega^*_{w,x}} i_j(\hat{\beta}_{j,*})$ to account for the fact that in $\mathbb{O}_{A,w}$, if firm w incentivizes firm x then the optimal incentive strategy is to also all firms in $\Omega^*_{w,x}$. The optimal incentive strategy for firm w is then the solution to $\mathbb{O}_{T,w}(B)$, for some $B \in K_w$, that maximizes firm w's rewards.

The exact algorithm for $\mathbb{O}_{A,w}$ requires the characterization of $\Omega^*_{w,y}$ for all $y \in Y_w$. This is achieved by solving $\mathbb{O}_{A,y}(B_y)$, where $B_y \in K_y$, while fixing d^*_y to v_y and defining the set $Y_y = Y_w \cap G_y$. Solving $\mathbb{O}_{A,w}$ is thus a recursive process where $\Omega^*_{w,y}$ is determined first by solving two-level problems for all sets $B_y \in K_y$ for each firm $y \in K_y$, before finally solving firm *w*'s incentive problem $\mathbb{O}_{A,w}$. Using the concepts presented above, we now define the exact algorithm for $\mathbb{O}_{A,w}$.

Exact algorithm for $\mathbb{O}_{A,w}$	
Input	
	Y_w
Initialize	
	$\hat{\beta}_{\text{INF,w,*}}, D^*, \tilde{\beta}^*_{\text{INF,w}}(D^*), d^*_w$
Execute	
1	For each $y \in Y_w$, set $d^*_y = v_y$ and determine $\Omega^*_{w,y}$ by solving $\mathbb{O}_{T,y}(B_y)$ for all $B_y \in K_y$.
2	Determine the set of firms $I^*_w \subset Y_w$ that maximizes firm *w*'s rewards by solving $\mathbb{O}_{T,w}(B)$ for all $B \in K_w$
Return I^*_w	

5.5 Numerical example

There are 18 firms in a network hierarchy, each of whom works on a critical component design for the main contractor, firm 1. The hierarchy consists of 4 levels. Firm 1 is on level 1, firms 2 to 4 are on level 2, firms 5 to 10 are on level 3 and firms 11 to 18 are on level 4. The set of supervising firms $W = \{1, 2, \ldots, 7, 9\}$, the set of subordinate firms $U = \{2, \ldots, 18\}$ and the set of firms that have no subordinates $U \setminus W = \{8, 10, 11, \ldots, 18\}$. Each supervising firm's set of immediate subordinates and set of control is presented in table 6. Values of the model parameters that we consider for this example is presented in table 7, the expected rewards for all firms, with and without incentives, along with the optimal strategies for supervising firms when verification activities are incentivized are presented in table 8, and the optimal verification strategies for all firms, with and without incentives, is graphically depicted in figure 7.

Table 3: Sets associated with supervising firms for the network model

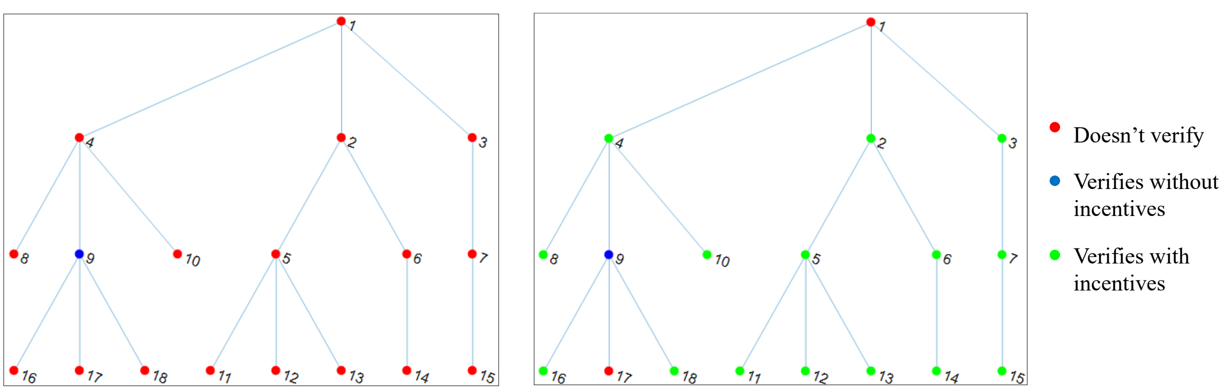
Supervising Firm	h_w	T_w	M_w
1	1	{2,3,4}	{2,3,4,5,6,7,8,9,10,16,17,18}
2	2	{5,6}	{5,6,11,12,13,14}
3	2	{7}	{7}
4	2	{8,9,10}	{8,9,10,16,17,18}
5	3	{11,12,13}	{11,12,13}
6	3	{14}	{14}
7	3	{15}	{15}
9	3	{16,17,18}	{16,17,18}

Table 4: Parameters for the network model

Firm	h_x	g_x	l_x	c_x	r_x	ε_x	θ_x	β_x
1	1	500,000	200,000	200,000	100,000	0.6	0.43	0.98
2	2	40,000	20,000	15,000	10,000	0.4	0.21	0.99
3	2	40,000	20,000	10,000	30,000	0.4	0.24	0.91
4	2	50,000	20,000	10,000	30,000	0.4	0.3	0.99
5	3	5,000	2,000	1,000	2,000	0.3	0.12	0.96
6	3	5,000	2,000	1,000	2,000	0.3	0.11	0.9
7	3	5,000	2,000	1,000	1,000	0.3	0.14	0.92
8	3	7,000	2,000	1,500	2,000	0.3	0.13	0.95
9	3	7,000	2,000	1,500	2,000	0.3	0.13	0.99
10	3	5,000	2,000	1,000	1,000	0.3	0.17	0.99
11	4	400	200	100	100	0.2	0.04	0.91
12	4	500	200	50	100	0.2	0.09	0.99
13	4	500	200	100	200	0.2	0.03	0.99
14	4	500	200	100	100	0.2	0.1	0.94
15	4	400	200	50	200	0.2	0.08	0.98
16	4	300	200	50	100	0.2	0.1	0.91
17	4	500	200	500	300	0.2	0.1	0.94
18	4	500	200	100	200	0.2	0.02	0.99

Table 5: Results for the network model example

Firm	Net expected reward when no firm incentivizes	Optimal incentive strategy (firms to incentivize)	Net expected reward after incentivization	Gain in expected rewards (%)
1	123,560	{2,3,4}	278,640	125.52
2	17,150	{5,6}	24,520	42.96
3	26,190	{7}	29,740	13.56
4	20,630	{8,10}	26,140	26.73
5	2,040	{11,12,13}	2,780	35.9
6	3,400	{14}	3,620	6.52
7	3,630	{15}	3,870	6.65
8	5,940	-	5,940	0
9	3,250	{16,18}	3,510	8.02
10	4,580	-	4,580	0
11	350	-	350	0
12	460	-	460	0
13	450	-	450	0
14	450	-	450	0
15	370	-	370	0
16	280	-	280	0
17	450	-	450	0
18	440	-	440	0



Optimal verification strategies without incentives

Optimal verification strategies with incentives

Figure 4: Graphical representation of the solutions for the network model example

Without incentives, no firm verifies its design except firm 9. For values of β_x considered for this example, the expected costs of verification outweigh the benefits of verification for all firms except firm 9 when verification activities are not incentivized. However, this is remedied when verification activities are incentivized. With incentives, all firms except firms 1 and 17 verify their design. Each firm in a supervising role incentivizes at least one subordinate firm. In addition, each supervising firm experiences a net gain in rewards when it chooses to incentivize the verification activities of one or more of its subordinates.

In both scenarios, the main contractor, firm 1, chooses not to verify its design. The reasons for the main contractor not verifying the system design in the two scenarios, however, are not the same. Without incentives, verification of the system design is too costly for the main contractor, and the main contractor prefers not to verify the system design even when none of its immediate subordinates verify their design. Whereas, with incentives, the main contractor chooses not to verify the system design due to its belief in the ideal state of the system design being higher than $\hat{\beta}_1^*$, which results from firms 2,3 and 4 verifying their design due to incentives from the main contractor. Of all the supervising firms, the main contractor benefits the most from verification activities being incentivized, with the expected reward of the main contractor more than doubling when verification activities are incentivized.

6 Model validity

We have developed a normative decision-theoretic model of verification in this paper. Our model was not developed using a dataset obtained from the industry and is theoretical in nature. Hence, a data-driven validation process is not applicable for our work. Instead, we validate our model with the intention of providing a potential user with more confidence in its applicability. In this regard, hypotheses validity and logical validity are two qualitative validation methods frequently used on decision-theoretic models [61, 62]. We discuss both below.

6.1 Hypothesis validity

Hypothesis validity checks if the model has adequately reproduced the connections between the elements of the *subject* being modeled [62, 63]. In the context of our model, the subject is the decision to verify (or not) a system design in a particular phase of its development. This decision affects a firm's confidence in the correctness of its system design as the design evolves over the development process, which is modeled as a belief distribution. Furthermore, this decision is governed by the costs associated with verification and the probability of a firm making a design error during the design activities. The inputs to our model are then the aforementioned parameters, with the outputs being the optimal decision, the firm's transformed belief in the correctness of its design and the expected cost of verification. Then, the connections between the elements of the subject, in the context of our model, are the relationships between the input parameters to the model and the output metrics observed from the model.

We say that the organization's confidence in the correctness of its design, represented as a belief in our model, binds all the parameters of our model. Our argument is as follows. The development process generates rich data in the form of design discussions, logs of activities, observations, and test results, for example. This rich data influences the organization's understanding of the state of its design. Since the true state of the design is unknown prior to verification, the organization's understanding of the state of its design is subjective. That is, the organization does not *know* the true state of its design but can be thought of as being confident in the correctness of the design. The organization will make verification decisions based on this confidence. Since the costs of verification are set by the organization's decision, it then follows that adequately modeling the organization's confidence in the correctness of its design activities is sufficient to connect our model input to its outputs.

There are two aspects to modeling the organization's confidence: 1) quantifying the confidence, and 2) modeling the change in this confidence. To quantify the organization's confidence in the correctness of its design activities, we use belief distributions. The organization's confidence is changed by the actions of the design activities. However, these activities have been abstracted away in our model. Thus, we need a parameter that adequately represents the way in which design activities vary the organization's belief in the correctness of its design. This function is accomplished by the probability of making a design error ε.

6.2 Logical validity

Logical validity checks if a model has been correctly converted into a numerical computer model that produces solutions [61]. There is no standard methodology for determining logical validity, but qualitative inspections have been used in the past [61]. To the best of our knowledge, the results of our model are numerically correct. However, we do contend that numerical accuracy does not necessarily imply applicability in reality. In this regard, our model makes two assumptions that leads to numerically correct but inapplicable results in those scenarios where the organization's baseline confidence in the correctness of its design maturity/capability is low throughout the design process: 1) the system design either is either faulty, or not and 2) when the system is verified, the belief in the correct state of the system design becomes absolute.

The two assumptions mentioned above, together, overlook the possibility of the system design being in more granular states during the design and verification process. Still, our model does derive a numerically correct strategy for those scenarios where the organization's baseline confidence in the correctness of its design activities is low – no verification in any phase but the last. This is so since our model suggests that even if the system is verified, the confidence of it being in the correct state will be low throughout the process, and hence it is best not to waste monetary resources on the same. However, in reality, the organization would prefer to verify its design if its baseline confidence in the correctness of its design activities is low. We conjecture that this issue can be resolved by expanding the size of the state space and by allowing a more granular increase in belief after verification activities.

7 Conclusion

Incentivizing verification activities in multi-firm design projects is a significant challenge for the main contractor. In this paper, we developed a belief-based modeling approach to derive optimal verification strategies in multi-firm scenarios, along with the incentives that can implement these strategies. The optimal incentives are a function of the subordinate firm's beliefs and the influence exerted by the subordinate firm on the supervising firm with respect to verification activities.

We presented the two-firm model of verification as a building block and then extended the results to three scenarios: 1) a two-level model and 2) the network model. For each scenario, we presented an exact algorithm that determines optimal verification and incentive strategies. The exact algorithms for the two-level and network models were both observed to be NP-hard. Numerical examples were presented for each scenario to illustrate the benefits of incentivizing verification activities.

In conclusion, our work lays a foundation for studying the problem of incentivizing verification in multi-firm scenarios. We focused on scenarios where the state space for each firm can be broadly categorized as either ideal or non-ideal and the decision space for each firm consisted of two possible actions. By deriving exact algorithms for all scenarios explored, we have also laid a foundation for the derivation of efficient heuristics that can determine near-optimal incentive and verification strategies for multi-firm projects with a large number of participating firms. Future work is suggested to incorporate explicit models of human and organizational behavior.

Appendix A

Proof of Theorem 1 We will prove Theorem 1 for $d_{\text{SUP}} = -v_{\text{SUP}}$. The proof for $d_{\text{SUP}} = v_{\text{SUP}}$ is similar. For any $A \in G_{c,k}$ and for any $k \in \{1,\ldots,|\overline{V}_c|\}$, we define $\overline{A} = \overline{V}_c \setminus A$. Since it is profitable for SUP to incentivize each firm in the set $\{x_1,\ldots,x_m\}$ individually, for $j \in \{1,\ldots,m\}$, we know

$$\text{E}(R_{\text{SUP}} \mid \ldots, Z_{I_c \cup \{x_j\}}, \ldots, -v_{\text{SUP}}) \geq \text{E}(R_{\text{SUP}} \mid \ldots, Z_{I_c}, \ldots, -v_{\text{SUP}})$$

$$\Rightarrow (g_{\text{SUP}} - l_{\text{SUP}})\beta_{\text{SUP}}(1 - \varepsilon_{\text{SUP}} + \theta_1) \prod_{u \in \overline{V}_c \setminus \{x_j\}} \hat{\beta}_u (1 - \hat{\beta}_{x_j}) \geq i_{x_j}(\hat{\beta}_{x_j}). \tag{24}$$

Let $j_1,\ldots,j_m$ be an arbitrary ordering of the set $\{x_1,\ldots,x_m\}$. This implies $j_1,\ldots,j_m \in \{x_1,\ldots,x_m\}$ and $j_1 \neq \ldots \neq j_m$. Using condition (24) we know

$$\text{E}(R_{\text{SUP}} \mid \ldots, Z_{I_c \cup \{j_1\} \cup \{j_2\}}, \ldots, d_{\text{SUP}}) - \text{E}(R_{\text{SUP}} \mid \ldots, Z_{I_c \cup \{j_1\}}, \ldots, d_{\text{SUP}})$$

$$= (g_{\text{SUP}} - l_{\text{SUP}})\beta_{\text{SUP}}(1 - \varepsilon_{\text{SUP}} + \theta_1) \prod_{u \in \overline{V}_c \setminus \{j_1\} \cup \{j_2\}} \hat{\beta}_u (1 - \hat{\beta}_{j_2}) - i_{j_2}(\hat{\beta}_{j_2}) \geq 0 \tag{25}$$

since $\prod_{u \in \overline{V}_c \setminus \{j_1\} \cup \{j_2\}} \hat{\beta}_u (1 - \hat{\beta}_{j_2}) \geq \prod_{u \in \overline{V}_c \setminus \{j_2\}} \hat{\beta}_u (1 - \hat{\beta}_{j_2})$. Similarly, for $2 < q \leq m$, we know

$$\text{E}(R_{\text{SUP}} \mid \ldots, Z_{I_c \cup \{j_1\} \cup \ldots \cup \{j_q\}}, \ldots, d_{\text{SUP}}) - \text{E}(R_{\text{SUP}} \mid \ldots, Z_{I_c \cup \{j_1\} \cup \ldots \cup \{j_{q-1}\}}, \ldots, d_{\text{SUP}})$$

$$= (g_{\text{SUP}} - l_{\text{SUP}})\beta_{\text{SUP}}(1-\varepsilon_{\text{SUP}} + \theta_1) \prod_{u \in \bar{V}_c \setminus \{j_1\} \cup \ldots \cup \{j_q\}} \hat{\beta}_u (1-\hat{\beta}_{j_q}) - i_{j_q}(\hat{\beta}_{j_q}) \geq 0 \qquad (26)$$

since $\prod_{u \in \bar{V}_c \setminus \{j_1\} \cup \ldots \cup \{j_q\}} \hat{\beta}_u (1-\hat{\beta}_{j_q}) \geq \prod_{u \in \bar{V}_c \setminus \{j_q\}} \hat{\beta}_u (1-\hat{\beta}_{j_q})$. The arbitrary ordering of $j_1, \ldots, j_m$ implies that for any $A \subseteq \{x_1, \ldots, x_m\}$, $\text{E}(R_{\text{SUP}} \mid \ldots, Z_{I_c \cup \{x_1, \ldots, x_m\}}, \ldots, d_{\text{SUP}}) \geq \text{E}(R_{\text{SUP}} \mid \ldots, Z_{I_c \cup A}, \ldots, d_{\text{SUP}})$ holds true ■

References

1. Salado, A., *Defining Better Test Strategies with Tradespace Exploration Techniques and Pareto Fronts: Application in an Industrial Project.* Systems Engineering, 2015. **18**(6): p. 639-658.
2. Engel, A., *Verification, Validation, and Testing of Engineered Systems*. Wiley Series on Systems Engineering and Management, ed. A.P. Sage. 2010, Hoboken, NJ: John Wiley & Sons, Inc.
3. Shabi, J., Y. Reich, and R. Diamant, *Planning the verification, validation, and testing process: a case study demonstrating a decision support model.* Journal of Engineering Design, 2017. **28**(3): p. 171-204.
4. Tahera, K., C. Earl, and C. Eckert, *A method for improving overlapping of testing and design.* IEEE Transactions on Engineering Management, 2017. **64**(2): p. 179-192.
5. Nagano, S., *Space systems verification program and management process: Importance of Implementing a Distributed-Verification Program with Standardized Modular-Management Process.* Systems Engineering, 2008. **11**(1): p. 27-38.
6. Collopy, P. *A research agenda for the coming renaissance in systems engineering*. in *50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*. 2012.
7. Ciucci, F., T. Honda, and M.C. Yang, *An information-passing strategy for achieving Pareto optimality in the design of complex systems.* Research in engineering design, 2012. **23**(1): p. 71-83.
8. Lewis, K. and F. Mistree, *Modeling interactions in multidisciplinary design: A game theoretic approach.* AIAA journal, 1997. **35**(8): p. 1387-1392.
9. Vermillion, S.D. and R.J. Malak. *A theoretical look at the impact of incentives on design problem effort provision*. in *ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. 2018. American Society of Mechanical Engineers Digital Collection.
10. Vermillion, S.D. and R.J. Malak, *A Game Theoretical Perspective on Incentivizing Collaboration in System Design*, in *Disciplinary Convergence in Systems Engineering Research*. 2018, Springer. p. 845-855.
11. Xiao, A., et al., *Collaborative multidisciplinary decision making using game theory and design capability indices.* Research in Engineering Design, 2005. **16**(1-2): p. 57-72.
12. Baiman, S., P.E. Fischer, and M.V. Rajan, *Information, contracting, and quality costs.* Management Science, 2000. **46**(6): p. 776-789.

13. Balachandran, K.R. and S. Radhakrishnan, *Quality implications of warranties in a supply chain.* Management science, 2005. **51**(8): p. 1266-1277.
14. Zhu, K., R.Q. Zhang, and F. Tsung, *Pushing quality improvement along supply chains.* Management science, 2007. **53**(3): p. 421-436.
15. Kulkarni, A.U., C. Wernz, and A. Salado, *Coordination of verification activities with incentives: a two-firm model.* Research in Engineering Design, 2021. **32**(1): p. 31-47.
16. Sentz, K. and S. Ferson, *Combination of evidence in Dempster-Shafer theory*. Vol. 4015. 2002: Citeseer.
17. Eifler, T., et al. *An assignment of methods to analyze uncertainty in different stages of the development process*. in *ASME 2010 International Mechanical Engineering Congress and Exposition*. 2010. American Society of Mechanical Engineers Digital Collection.
18. Morse, E., et al., *Tolerancing: Managing uncertainty from conceptual design to final product.* CIRP Annals, 2018. **67**(2): p. 695-717.
19. Wynn, D.C., K. Grebici, and P.J. Clarkson, *Modelling the evolution of uncertainty levels during design.* International Journal on Interactive Design and Manufacturing (IJIDeM), 2011. **5**(3): p. 187.
20. Tosney, W.F. and S. Pavlica. *Satellite verification planning: Best practices and pitfalls related to testing*. in *Environmental Testing for Space Programmes*. 2004.
21. Barad, M. and A. Engel, *Optimizing VVT strategies: a decomposition approach.* Journal of the Operational Research Society, 2006. **57**(8): p. 965-974.
22. Engel, A., *Verification, validation, and testing of engineered systems*. Vol. 73. 2010: John Wiley & Sons.
23. Engel, A. and M. Barad, *A methodology for modeling VVT risks and costs.* Systems Engineering, 2003. **6**(3): p. 135-151.
24. Engel, A. and M. Last, *Modeling software testing costs and risks using fuzzy logic paradigm.* Journal of Systems and Software, 2007. **80**(6): p. 817-835.
25. Goel, A.L. and K. Okumoto, *Time-dependent error-detection rate model for software reliability and other performance measures.* IEEE Transactions on Reliability, 1979. **28**(3): p. 206-211.
26. Ha, A.Y. and E.L. Porteus, *Optimal timing of reviews in concurrent design for manufacturability.* Management Science, 1995. **41**(9): p. 1431-1447.
27. Hossain, S.A. and R.C. Dahiya, *Estimating the parameters of a non-homogeneous Poisson-process model for software reliability.* IEEE Transactions on Reliability, 1993. **42**(4): p. 604-612.
28. Loch, C.H., C. Terwiesch, and S. Thomke, *Parallel and sequential testing of design alternatives.* Management Science, 2001. **47**(5): p. 663-678.
29. McGarry, F. and G. Page, *Performance evaluation of an independent software verification and integration process.* NASA Goddard, Greenbelt, MD, SEL Sill 0, 1982.
30. Powell, P.B., *Software validation, verification, and testing technique and tool reference guide.* 1982.
31. Shabi, J. and Y. Reich, *Developing an analytical model for planning systems verification, validation and testing processes.* Advanced Engineering Informatics, 2012. **26**(2): p. 429-438.
32. Tahera, K., *The role of testing in engineering product development processes*. 2014, The Open University.

33. Tahera, K., C.F. Earl, and C.M. Eckert, *Integrating virtual and physical testing to accelerate the engineering product development process.* IJITM, 2014. **13**(2/3): p. 154-175.
34. Thomke, S. and D.E. Bell, *Sequential testing in product development.* Management Science, 2001. **47**(2): p. 308-323.
35. Wallace, D.R. and R.U. Fujii, *Software verification and validation: an overview.* Ieee Software, 1989. **6**(3): p. 10-17.
36. Yamada, S., T. Ichimori, and M. Nishiwaki, *Optimal allocation policies for testing-resource based on a software reliability growth model.* Mathematical and Computer Modelling, 1995. **22**(10-12): p. 295-301.
37. Yamada, S., M. Ohba, and S. Osaki, *S-shaped reliability growth modeling for software error detection.* IEEE Transactions on Reliability, 1983. **32**(5): p. 475-484.
38. Kulkarni, A.U., et al. *Is Verifying Frequently an Optimal Strategy? A Belief-Based Model of Verification*. in *ASME 2020 International Design Engineering Technical Conference & Computers and Information in Engineering Conference (IDETC/CIE 2020)*. 2020. St. Louis, MO, (USA).
39. Kulkarni, A.U., et al., *An evaluation of the optimality of frequent verification for vertically integrated systems.* Systems Engineering, 2021. **24**(1): p. 17-33.
40. Farkhondehmaal, F. and A. Salado, *Efficient population of the verification tradespace using Bayesian inference.* IEEE Systems Journal, 2019. **14**(3): p. 3225-3232.
41. Salado, A. and H. Kannan, *Properties of the Utility of Verification*, in *IEEE International Symposium in Systems Engineering*. 2018: Rome, Italy.
42. Salado, A. and H. Kannan, *A mathematical model of verification strategies.* Systems Engineering, 2018. **21**(6): p. 593-608.
43. Salado, A. and H. Kannan. *Properties of the Utility of Verification*. in *2018 IEEE International Systems Engineering Symposium (ISSE)*. 2018. IEEE.
44. Salado, A. and H. Kannan, *Elemental patterns of verification strategies.* Systems Engineering, 2019. **22**(5): p. 370-388.
45. Salado, A., H. Kannan, and F. Farkhondehmaal, *Capturing the Information Dependencies of Verification Activities with Bayesian Networks*, in *Conference on Systems Engineering Research (CSER)*. 2018: Charlottesville, VA, USA.
46. Xu, P. and A. Salado. *A Concept for Set-based Design of Verification Strategies*. in *INCOSE International Symposium*. 2019. Wiley Online Library.
47. Xu, P., A. Salado, and G. Xie. *A Reinforcement Learning Approach to Design Verification Strategies of Engineered Systems*. in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2020. IEEE.
48. Xu, P. and A. Salado. *Towards the Dynamic Contracting of Verification Activities with Set-Based Design: An Initial Model of Rework*. in *Sixteenth Annual Acquisition Research Symposium*. 2019. Monterey, CA, USA.
49. Xu, P. and A. Salado. *A Concept for Set-based Design of Verification Strategies*. in *INCOSE International Symposium*. 2019. Orlando, FL, USA.
50. Kulkarni, A.U., et al., *An evaluation of the optimality of frequent verification for vertically integrated systems.* Systems Engineering, 2020. **24**(1): p. 17-33.
51. Kulkarni, A.U. and C. Wernz, *Optimal incentives for teams: a multiscale decision theory approach.* Annals of Operations Research, 2020. **288**: p. 307-329.

52. Kulkarni, A.U., et al. *Is Verifying Frequently an Optimal Strategy? A Belief-Based Model of Verification*. in *ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. 2020.
53. Emons, W., *Warranties, moral hazard, and the lemons problem.* Journal of Economic Theory, 1988. **46**(1): p. 16-33.
54. Reyniers, D.J., *Supplier-customer interaction in quality control.* Annals of Operations Research, 1992. **34**(1): p. 307-330.
55. Reyniers, D.J. and C.S. Tapiero, *Contract design and the control of quality in a conflictual environment.* European journal of operational research, 1995. **82**(2): p. 373-382.
56. Starbird, S.A., *Penalties, rewards, and inspection: provisions for quality in supply chain contracts.* Journal of the Operational Research Society, 2001. **52**(1): p. 109-115.
57. Wernz, C., *Multiscale Decision Making: Bridging Temporal and Organizational Scales in Hierarchical Systems*, in *Mechanical and Industrial Engineering*. 2008, Dissertation, University of Massachusetts Amherst.
58. Howard, R.A. and A.E. Abbas, *Foundations of Decision Analysis*. 1st Edition ed. 2015: Pearson.
59. Xu, P. and A. Salado, *Towards the Dynamic Contracting of Verification Activities With Set-Based Design: An Initial Model of Rework.* 2019.
60. Wang, X., W. Wu, and D. Zhu. *An approximation algorithm for nonlinear 0-1 integer programming problems*. in *Conference on Computer and Management (CAMAN), 2011 International* 2011. IEEE.
61. Gass, S.I., *Decision-aiding models: validation, assessment, and related issues for policy analysis.* Operations Research, 1983. **31**(4): p. 603-631.
62. Mihram, G.A., *Some practical aspects of the verification and validation of simulation models.* Journal of the Operational Research Society, 1972. **23**(1): p. 17-29.
63. McCarl, B.A., *Model validation: an overview with some emphasis on risk models.* Review of Marketing and Agricultural Economics, 1984. **52**(430-2016-31544): p. 153-173.

Biographies

Dr. Aditya U. Kulkarni was a Postdoctoral Associate in the Grado Department of Industrial and Systems Engineering at Virginia Tech at the time of writing this article. He obtained his Ph.D. in Industrial Engineering from Virginia Tech in 2018. He holds an M.S. in Industrial Engineering from Virginia Tech (2012) and a B.Tech in Mechanical Engineering from National Institute of Technology, Karnataka (2010). Dr. Kulkarni's research focuses on leveraging stochastic modeling and game theory to solve research problems in in Systems Engineering. He's currently working on determining optimal verification strategies in systems engineering projects.

Dr. Alejandro Salado has over 15 years of experience as a systems engineer, consultant, researcher, and instructor. He is currently an associate professor of systems engineering with the Department of Systems and Industrial Engineering at the University of Arizona. In addition, he provides part-time consulting in areas related to enterprise transformation, cultural change of technical teams, systems engineering, and engineering strategy. Alejandro conducts research in problem formulation, design of verification and validation strategies, model-based systems engineering, and engineering education. Before joining academia, he held positions as systems engineer, chief architect, and chief systems engineer in manned and unmanned space systems of up to $1B in development cost. He has published over 100 technical papers, and his research has received federal funding from the National Science Foundation (NSF), the Naval Surface Warfare Command (NSWC), the Naval Air System Command (NAVAIR), and the Office of Naval Research (ONR), among others. He is a recipient of the NSF CAREER Award, the International Fulbright Science and Technology Award, the Omega Alpha Association's Exemplary Dissertation Award, and several best paper awards. Dr. Salado holds a BS/MS in electrical and computer engineering from the Polytechnic University of Valencia, a MS in project management and a MS in electronics engineering from the Polytechnic University of Catalonia, the SpaceTech MEng in space systems engineering from the Technical University of Delft, and a PhD in systems engineering from the Stevens Institute of Technology. Alejandro is a member of INCOSE and a senior member of IEEE and AIAA.

Dr. Christian Wernz is a Senior Data Scientist at the University of Virginia Health System (UVA Health). He was an Associate Professor in the Department of Health Administration at Virginia Commonwealth University (VCU) at the time of performing this research, and previously, he worked as a faculty in Industrial and Systems Engineering at Virginia. Dr. Wernz received his doctorate in Industrial Engineering and Operations Research from the University of Massachusetts Amherst and his bachelor's and master's degrees in Business Engineering from the Karlsruhe Institute of Technology (KIT), Germany. In his research, Dr. Wernz models and analyzes decision-making challenges in complex systems, with a particular interest in systems engineering and healthcare. His methodological expertise lies in multiscale decision theory, game theory, Markov decision processes, decision and data analytics, simulation, and healthcare delivery science. His research has been funded by the National Science Foundation (NSF), the Agency for Healthcare Research & Quality (AHRQ), the Harvey L. Neiman Health Policy Institute, Rolls-Royce, Dell and other industry partners.