Expert Systems With Applications 215 (2023) 119340

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Agency theory: Forecasting agent remuneration at insurance companies

Fernando Turrado García [a], Ana Lucila Sandoval Orozco [a], M. Pilar García Pineda [b], Luis Javier García Villalba [a,*]

[a] Group of Analysis, Security and Systems (GASS), Department of Software Engineering and Artificial Intelligence (DISIA), Faculty of Computer Science and Engineering, Office 431, Universidad Complutense de Madrid (UCM), Calle Profesor José García Santesmases 9, Ciudad Universitaria, 28040 Madrid, Spain

[b] Group of Analysis, Security and Systems (GASS), Department of Financial and Actuarial Economics & Statistics, Faculty of Economics and Business Administration, Universidad Complutense de Madrid (UCM), Campus Somosaguas, 28223 Pozuelo de Alarcón, Spain

ARTICLE INFO

Keywords:
Agent theory
Optimal contract
Supervised learning
Support vector machines
Montecarlo methods

ABSTRACT

The principal–agent problem occurs when one entity (the "agent"), is able to make decisions and/or take actions on behalf of another person or entity (the "principal"). The agent earnings are regulated under a contract designed by the principal. Under the principal's point of view, the main goal while designing said contract (and the payment rules incorporated on it) is to align the actions made by the agent to the principal's own goals. So, in this paper we will define a method that will allow the principal to forecast the remuneration obtained by the agent under an established contract in the insurance sector.

1. Introduction

In economic studies, the analysis of situations in which an entity (the agent) makes decisions on behalf of another (the principal) according to their own moral principles or interests is known as the principal–agent problem. In addition to this problem, the principal only has partial or inaccurate information about the activities of the agent he hires. In general terms, the rules that define the relationship between them are regulated in a contract that contains both the agent's obligations (tasks entrusted by the principal) and the rules by which he will be paid (payment commitments acquired by the principal).

This problem has been studied from different perspectives, such as the one described in the work of Holmstrom and Milgrom (1987) in which they analysed the definition of objectives for an agent with absolute risk aversion. Another example of this type of study can be found in Schättler and Sung (1993) where the authors – Schattler and Sung – applied Martingale methods (developed for the control of stochastic processes) to represent the agent's salary. Another work that belongs to this field of study (but focuses on the study of the establishment of dynamic contracts when there is uncertainty about the performance of the agent), can be found in the work of Prat and Jovanovic (2014). In Williams (2015) a similar problem is studied in which production is considered linear and utility exponential in which there are hidden states and actions. A bayesian aproach to solve this same problem can be found in the work of Castiglioni et al. (2022).

Some examples of more recent work can be found in Chen et al. (2017) and Lai et al. (2021) or Luo and Saigal (2021). In the paper written by Chen et al. (2017), an approximation of this problem is made when the agents are dedicated to the sale of the products manufactured by the principal in the supply chain. Lai et al. (2021) propose another alternative method for determining the optimal contract when information regarding the agent's performance is unknown. In Luo and Saigal (2021) a set of problems of this type is modelled as a Hamilton–Jacobi-Bellman type equation with equilibrium constraints. Lastly, in Castiglioni et al. (2022) the authors tackle a principal–agent problem using a Bayesian approach to calculate optimal linear or tractable contracts.

In this paper, the principal represents an insurance company that arranges a set of insurance products from different branches. Regarding the figure of the agent, this is an insurance mediator in charge of arranging the insurance contracts of the company, but the insurance mediator works on their own behalf under another contract (like a vendor service agreement). This contract usually contains legal terms for the following topics: remuneration rules and some review periods when the agent's performance is reviewed by the principal (for example, monthly or quarterly).

Remuneration rules, settled by the principal to remunerate the agent, are grouped into two main concepts: commissions and incentives. Commissions are understood as those perceptions received by the agent based on a transaction (for example, the sign of an insurance contract or policy). Incentives are understood as those perceptions derived from the fulfilment of certain objectives based on sets of transactions,

* Corresponding author.
E-mail addresses: fernturr@ucm.es (F. Turrado García), asandov@ucm.es (A.L. Sandoval Orozco), mpigarci@ucm.es (M.P. García Pineda), javierv@ucm.es (L.J. García Villalba).

https://doi.org/10.1016/j.eswa.2022.119340
Received 31 July 2022; Received in revised form 31 October 2022; Accepted 20 November 2022
Available online 24 November 2022
0957-4174/© 2022 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

such as, for example, selling a predetermined number of policies in the same month.

Thus, the relationship between both entities is defined as follows: the principal expects the agent to dedicate their effort (productive capacity) to arrange the most interesting insurance contracts or policies under their business strategy (which may be known by the agent). On their part, the agent tries to maximize their remuneration by arranging the principal's contracts subject to their productive capacity (maximum level of effort that he can perform per review period).

Asymmetry, difference of interests or moral conflict arises from the practical application of the remuneration rules defined in the contract, since the combination of arranged contracts that maximizes the remuneration obtained by the agent may not be the one expected by the principal.

1.1. Original contributions

Recent studies about the agency problem focus on analysing said problem from the perspective of business management or management science, as seen in different works such as those of Huang et al. (2022) where the authors found that how the agency problem affects a firm's cash holdings in China, or in the research of Turnbull et al. (2022) focuses on mitigating the agency costs in the housing market. In different sort of study, the paper written by Chod and Lyandres (2021) studies the impact of the agency problem on the financing of entrepreneurial ventures via crypto tokens.

Taking a different point of view, closer to the computer science and artificial intelligence, this problem can be categorized (according to the criteria defined by Albrecht and Stone (2018)) as type of problem where: the agent makes deterministic choices with limited or partial information. The agent can also change their behaviour based on the outcomes of prior actions or decisions, with some key factors which are unknown to either the principal (such as the internal effort function of the agent) or the agent (the ex-ante premium amount involved in selling an insurance policy); the action choices are independent from one another and conflicting goals arise. However, when looking at the main differences between this paper and that of Castiglioni et al. (2022) (which is taken as an example). In this paper, the results are calculated based on stochastic variables, and the principal is unaware of the action costs, or, in essence, the agent's effort function.

The study of the agency problem is usually carried out from either an economic or business management approach, and hence, the number of publications focused on describing calculation methods for this kind of problem is very limited; this is further aggravated when the particularities of certain domains (e.g., the insurance domain) are considered.

This paper proposes a method that allows the principal to estimate the optimal combination of contracts to maximize the agent's profit. Through these estimates, the principal will be able to suggest the necessary modifications to the remuneration rules (hence in the contract) and thus be able to align their interests with those of the agent.

The proposed method for modelling the agent's behaviour combines both combinatorial optimization and supervised learning techniques; on one hand, optimization is used for finding solutions, and supervised learning techniques are used for narrowing the feasible solutions to be evaluated. These optimization techniques guarantee that the optimal solution found will represent a value that is close to the real maximum.

However, this maximum can be reached through a combination of contracts that does not correspond to the agent's profile. As an example, we will assume that the maximum quoted above is reached by signing 4 type A contracts and 6 type B contracts. In this specific case, if the agent has never signed type B contracts, the global optimal solution does not present a realistic solution for the behaviour of the agent based on its historical data.

To minimize this effect, we propose using a classifier based on supervised learning, which would allow a comparison between the solutions suggested by the combinatorial optimization algorithm and the recorded historical data, thus disregarding those that are not similar enough. The proposed classifier is incorporated as an additional constraint in the optimization problem.

Lastly, this method also sorts out one of the open problems listed by Albrecht and Stone (2018) on the survey: Modelling with action duration. In this case, the actions have an instant effect (the commission associated to each sale) and a delayed effect (the remuneration obtained with the incentives) that is only visible by both entities at the end of the review period.

2. Material and methods

2.1. Formal problem definition

First, we will proceed to the formal definition of the problem under study, the estimation of the impact of the changes which must be introduced by the principal to make the agent's commercial effort focus, and hence, assess the combination of insurance contracts desired from the principal.

In our study, we assume that the principal commercializes a fixed set of insurance contracts.

R will be said set of insurance contract types, hereby defined as

$$R = \{r_i, i \in [1, N]\} \tag{1}$$

For each insurance contract type there are several features that can be used (from the principal point of view) to measure the agent's performance on its mediation labour (for example, warranties or premium amounts). However, we will only consider three concepts: number of policies arranged grouped by type, total amount of premium and number of insured people (in addition to the policy holder).

Then, we will define ”transaction” as the arrangement of an insurance contract (or policy), which entails obtaining a specific amount of premium and number of insured people, and both amounts being related (the more insured people, the higher amount of premium). Similarly, the effort required to arrange the policy varies based on the concepts described above because arranging certain types of policies will require greater amounts of effort from the agent.

So, in formal language, one transaction is defined as a tuple T_j where:

$$T_j = (r_j, p_j, a_j, e_j) \tag{2}$$

and r_j represents the policy type, p_j represents the premium, a_j represents the number of insured people and finally e_j represents the effort needed to arrange the policy.

However, for insurance contracts, both the premium and the number of insured people are unknown by the agent when he decides to start arranging a policy. This situation does not happen in other sectors, such as the retail, where the goods' value is predetermined in advance and the agent is fully aware of it. So, to define the prior concepts in formally, we will assume that said concepts follow a probability distribution function that may vary from each insurance contract type.

Therefore, those concepts must be rewritten in the following way:

- Let A_j be the probability distribution function for the number of insured people related to the j type of insurance contract.
- Let P_j be the probability distribution function for the premium related to the j type of insurance contract and the number of insured people.
- Let E_j be the probability distribution function for effort related to the j type of insurance contract, the number of insured people and the premium.

Finally, the T_i transaction becomes

$$T_i = (r_i, P_{r_i}(A_{r_i}), A_{r_i}, E_{r_i}(P_{r_i}(A_{r_i}), A_{r_i})) \quad (3)$$

From now, T will be denoted as the set of transactions arranged by the agent in a review period.

Once the transaction concept has been defined, we will proceed to define the concepts related to the agent's remuneration: commissions and incentives. As stated before, commissions are calculated over single transactions, while incentives are calculated – usually at the end of a review period – over sets of transactions.

As indicated in the previous section, commissions are calculated on the policy type, premium and number of insured people of a transaction. So an abstract definition for a commission rule would be: $c_k(t_i) = C_k(r_i, a_i, p_i)$ where C_k is an arbitrary function. However, in the scope of this paper we will assume a finite quantity (for example M) commissioning rules that are defined by linear or piece-wise linear functions based on the previous concepts. C will be denoted as the set of commissioning rules.

Regarding the other kind of remuneration (the incentives set for the period under review), we will formally define them as conditional rules such as the following: if the total contracted premium exceeds X, a compensation of Y is obtained in that period. Thus, we can formally define each incentive as follows:

$$i_l(T) = \begin{cases} I_l(T) & \text{if } IC_l(T) \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where IC_l represents the condition required to obtain the incentive and I_l represents the compensation obtained.

As in the case of commissions, in this paper we will assume that, on one hand, the condition rule (IC_l) is defined by a linear inequality on the transaction concepts, and, in the other hand, the compensation function (I_l) is a linear function. I will be denoted as the set of incentive rules.

Putting both concepts together, the remuneration obtained by the agent in the review period is calculated as:

$$\sum_{t_i \in T} \sum_{c_j \in C} c_j(t_i) + \sum_{i_k \in I} i_k(T) \quad (5)$$

Finally, it is necessary to define the constraint associated to the effort that the agent must make to arrange the set of transactions T. Thus, if we take into account the fact that the agent can only develop an amount E of maximum effort during the review period, we obtain the constraint by which the sum of the effort required for the set of transactions cannot exceed said that maximum:

$$\sum_{t_i \in T} e_i \leq E \quad (6)$$

Once this has all been sorted, the general formulation of the problem is written as:

$$\begin{aligned} Maximize & \sum_{t_i \in T} \sum_{c_j \in C} c_j(t_i) + \sum_{i_k \in I} i_k(T) \\ & \sum_{t_i \in T} e_i \leq E \end{aligned} \quad (7)$$

From the agent's point of view, their main goal is to find the optimal solution that maximizes the remuneration subject to the effort constraint. However, the principal must solve the same problem by looking for the optimal solution that best suits their own interests by altering the parameters of the remuneration rules (commissions and incentives). Also, the following considerations must be taken into account:

- The remuneration rules (commissions and incentives) are known by both entities. However, only the principal is entitled to change them.
- The effort needed for the arrangement of each transaction is known only by the agent. Because of this, the principal must estimate these coefficients to be able to solve the problem.
- The presence of the probability functions (associated with both the premium and the insured people) adds a stochastic nature to the problem under study. In this paper, we will assume that these functions are known by both entities.

2.2. Stochastic knapsack problem

In this way, the defined problem results in a variation of the knapsack problem defined by Dantzig (1957), to which a stochastic nature is incorporated, derived from the fact that both the value of the items and their weight are determined in a non-deterministic way. In literature, starting from the original definition of 1957, different extents to this class of stochastic knapsack problems (SKP) with random rewards have been identified. For example, in the one described as static SKP (see Henig (1990) and Morton and Wood (1998)), all the items are known from the start and the aim is to identify the combination that maximizes the probability of reaching a predetermined level of the objective function, assuming that the value of items is normally distributed. On the other hand, in Dynamic SKP (see Slyke and Young (2000)), items with random characteristics (value and/or weight) appear in the system as time progresses and it must be decided whether to accept the item once its value and weight has been known.

In this case, the problem under study differs from the previous ones due to the two interrelated characteristics that will be developed in the following paragraphs. First, the value of the item (premium) is not known by the agent until the final steps during the arrangement of the transaction. Despite this, the agent may know certain basic statistical properties (mean, mode, etc.) calculated based on previous transactions arranged previously. This partial knowledge allows the agent to develop certain strategies or predictions when deciding which policy to incorporate next. Second, and given that the effort required to arrange a given transaction varies depending on its characteristics, the agent must estimate which is the next best action on behalf on their own interests.

This way, the problem within the insurance sector differs from the static problem since, although the agent knows the insurance contract types he does not know the key characteristics associated with them prior to the policy arrangement. It also differs from the dynamic problem, since the policies do not appear randomly in the system, but rather the agent looks for those that optimizes their interest. On the other hand, derived from its choice, there is no decision whether to accept the item or whether not to, since there is a lack of penalties, the objective function always increases with each transaction.

Even though the fact that in its original formulation the number of policies must be modelled using integer variables, in this case we will define them as continuous variables to incorporate a new concept to the problem. The new concept consists of modelling the cases in which the commercial procedures associated with the management of a policy are carried out in several different review periods (for example, when 70% of the effort is devoted to period N and the remaining 30% in period N+1). Thus, when evaluating the continuous variables, the following consideration will be taken into account: the integer part of them represents the number of policies managed in said period and the remaining part represents the percentage of effort dedicated to the management of a policy that will be formalized in the future.

3. Theory/calculation

3.1. Proposed algorithm

In this section we will outline the algorithm proposed for the resolution of the problem described before from the principal point of view. This algorithm is based on the combination of three known techniques: mathematical programming, supervised learning and Montecarlo simulation.

First, and because the principal does not know the effort function of the agent (defined in Eq. (8)), our proposal is to train a machine learning classifier that can differentiate a set of transactions made by said agent from other sets of transactions. This training can be done using the historical set of transactions recorded by the principal. To guarantee that the mathematical programming algorithm finds at least an optimum, we will also introduce an additional restriction that limits the maximum number of policies that the agent can manage in a review period.

So, the problem is reformulated as:

$$
\begin{aligned}
&Maximize \sum_{t_i \in T} \sum_{c_j \in C} c_j(t_i) + \sum_{I_k \in I} I_k(T) \\
&\quad\quad\quad \mathbf{card}(T) \leq K \\
&\quad\quad\quad classifier(T) \geq \delta
\end{aligned}
\tag{8}
$$

To solve this, we will use the COBYLA algorithm proposed by Powell (1994) and revised at Powell (1998) because it is an algorithm for derivative-free optimization with nonlinear inequality constraints.

It is now crucial to address the stochastic nature of the problem generated by the presence of probability functions that determine key values (such as premium or insured people) for each transaction. So we propose to apply a Montecarlo approach by generating a random value (from each given probability distribution) every time the COBYLA algorithm needs a fixed value from one of those decision variables. With this approach, the resolution of many "fixed value problems" is required in order to give a stable solution.

The proposed method is defined at algorithm 1.

Algorithm 1 Algorithm overview

```
Require: Let N be the number of scenarios to simulate.
Require: Let H be the historical set of transactions recorded by the principal.
Require: Let δ be the confidence threshold for the classifier and K the empirical maximum
    of policies that can be arranged in a single review period.
Train a classifier responsible for deciding if a given set of transactions could have been
    made by the agent or not.
for scenario = 1, 2, …, N do
    Solve the current scenario with the adapted COBYLA algorithm for the δ and K
    values.
    Keep the results obtained after solving the problem above.
end for
Aggregate the N results in order to provide the final solution.
```

In the following sections, we will provide further details on the following key steps:

- Addressing the internal agent effort function.
- Montecarlo approach to bypass the stochastic nature.
- Summarizing the results.

3.2. Addressing the internal agent effort function

In this section we will provide further details regarding the necessary steps to train the classifier. This procedure has the following requirements:

- The principal has enough historical data obtained from the transactions arranged by the agent or different ones that have the same relationship with the principal.
- Only summarized or grouped data for each policy type is required to train the classifier (i.e. the order in which the agent has arranged the insurance contracts is not relevant).
- If needed, it is possible to augment the sample data using an algorithm similar to the one described at Turrado García et al. (2012) where the authors added new entries to an existing sample by mutating some observations.

So let be $E_s(T)$ the target feature defined as $E_s(T) = 1$ if the agent arranged the given set of transactions and $E_s(T) = 0$ otherwise. Regarding the predicting features, they where defined for each policy type as the amount of policies, the sum of insured people and the sum of premium of the transactions contained at each set.

For the supervised learning algorithm, we chose Support Vector Machines (from now on to be referred to as SVM) because it is a well known algorithm that was successfully applied to classification problems like the work done in Turrado García et al. (2012). It was also used or evolved in recent papers like the work of Dudzik et al. (2021) and Ma et al. (2021) or Lee et al. (2020).

Also, as shown in Turrado García et al. (2019) where the authors made a comparison between several supervised learning methods, the support vector machines performed better than other algorithms when used in combination with other predictors used to forecast some of the predicting features.

3.3. Classifier training description

Now, we will describe the procedure used to build the training set for the supervised learning classifier mentioned beforehand. The main concerns that drove the design of the procedure were:

- Real transactions made by the agent had to label led as belonging to the positive class.
- Due to the presence of stochastic features (the premium amount and insured people) in each transaction, additional random samples (using the related known probability distributions) had to be added to the positive class.
- The negative class for the classifier had to be randomly generated using the related known probability distributions.

The procedure used for generating the training set is shown at algorithm 2.

Algorithm 2 Procedure for building the training set

```
Require: Let TrainingSet be the training set.
Require: Let PO_i be the amount of policies of contract type i.
Require: Let PR_i be the amount of premium related to PO_i.
Require: Let IP_i be the amount of insured people related to PO_i.
Require: Let label be the classifier label (positive or negative).
Require: Let a sample be the tuple formed by a label and PO_i, PR_i and IP_i for each
    contract type.
Require: Let RealTransactionsSet be the set of real transactions made by the agent grouped
    by review period.
Require: Let M the number of positive samples to be generated for each tuple.
Require: Let NegSamples the number of negative samples to be generated.
TrainingSet ← RealTransactionsSet with all samples labelled as positive.
PositiveCases ← the set of unique tuples (PO_1, .., PO_i, .., PO_N)
for t = 1, 2, …, card(PositiveCases) do                    ▷ Random positive samples
    tuple ← (PO^t_1, …, PO^t_i, …, PO^t_N)
    for m = 1, 2, …,M do                                   ▷ M positive samples for each tuple
        for i = 1, 2, …,N do
            Generate PO^t_i random values for PR_i and IP_i.
            PR^t_i ← Σ_{j=1}^{PO^t_i} ( random PR^j_i )
            IP^t_i ← Σ_{j=1}^{PO^t_i} ( random IP^j_i )
        end for
        Add the m sample to TrainingSet as positive.
    end for
end for
MaxPO_i ← Max_{RealTransactionsSet}(PO_i) ∀ i ∈ {1, …, N}.
p ← 0
while p ≤ NegSamples do                                    ▷ Random negative samples
    Generate PO_i from Uniform(0, MaxPO_i) ∀ i ∈ {1, …, N}.
    if (PO_1, …, PO_i, …, PO_N) ∉ PositiveCases then
        for i = 1, 2, …,N do
            Generate PO^t_i random values for PR_i and IP_i.
            PR^t_i ← Σ_{j=1}^{PO^t_i} ( random PR^j_i )
            IP^t_i ← Σ_{j=1}^{PO^t_i} ( random IP^j_i )
        end for
        Add the p sample to TrainingSet as negative.
        p ← p + 1
    end if
end while
```

3.4. Montecarlo approach to bypass the stochastic nature

In this section we will describe how to use random sampling (a Montecarlo based method) to manage the stochastic variables features involved in a transaction. Those variables were separated into two:

- Decision variables whose values were determined by the COBYLA algorithm (for each policy type, the number of policies arranged).
- Stochastic variables related to each transaction: premium and insured people. The values for these variables are randomly generated using the related known probability distributions.

The key concepts behind this strategy are:

- The algorithm stores a list of transactions randomly generated for each policy type as an internal state. Those lists were ordered – following the agent's point of view – from better (higher) to worse values of premium and insured people.
- The transactions are randomly generated only when needed by the adapted COBYLA algorithm. Let us illustrate this with an example: consider a situation where COBYLA algorithm is evaluating a set of transactions that contains 8 policies of type i but the internal state only contains 6. At this situation, the algorithm must randomly generate 2 transactions of policy type i.
- The transactions were not removed from the internal state.
- When evaluating the objective function or the constraints, the adapted COBYLA algorithm only takes into account the correct number of policies, processing the lists in the most beneficial way for the agent (i.e., ordered from better to worse).

The proposed pseudo-code for the randomly transaction generation can be found in algorithm 3.

Algorithm 3 Random generation of transactions

Require: Let Trs be the set of ordered lists of randomly generated transactions for all the policy types.
Require: Let X the solution proposed by the COBYLA algorithm at a given iteration.
for $i = 1, 2, \ldots, N$ **do**
  Let Trs_i be the ordered list of randomly generated transactions for policy type i.
  Let X_i be the amount of transactions being evaluated at a given iteration of the COBYLA algorithm.
  if $X_i \geq \mathbf{card}(Trs_i)$ **then**
    $transactions \leftarrow X_i - \mathbf{card}(Trs_i)$
    for $temporalTransaction = 1, 2, \ldots, transactions$ **do**
      Generate a random transaction (Trs_{i_k}) for policy type i using the known probability distribution functions for insured people and premium.
      Add the generated transaction Trs_{i_k} to Trs_i
    end for
  end if
end for
Evaluate the constraints and objective function using the generated transactions set.

Although there were other strategies that could be executed while ordering the above mentioned lists, we chose this one due to its similarity to the Monte Carlo Tree Search method proposed by Coulom (2009) that focuses on the analysis of the most promising moves, expanding the search tree based on random sampling of the search space. One recent application of this method can be found at Shao et al. (2021) where the authors used it as a hierarchical task network planning method. So, the algorithm proposed will solve N problems starting from an empty state and giving, as a result, the reached optimum (set of transactions and objective function value).

3.5. Summarizing the results

In this section we will describe how we aggregated the different optimum solutions to obtain the probability function that represented the expected earnings for the agent.

Due to the stochastic nature of the problem, defining the solution as the maximum profit obtained by the agent, despite being mathematically feasible, is not realistic in practice (it is highly unlikely that an agent will arrange several policies with the maximum possible amount of premium and insured people at the same review period). In addition, although the grouping could also be performed on the real optimal value, a certain level of rounding (for example to the hundred) can be assumed without losing generality in the process. This is due to the differences in cents or euros when defining the agent's remuneration should not be considered relevant for the calculation. The proposed pseudo-code for the aggregation algorithm is shown at algorithm 4.

Table 1
Remuneration bins.

Bin	Rounded remuneration	Count	Probability
1	1200	25	0.000251
2	1250	408	0.0041
3	1300	1989	0.01999
4	1350	4028	0.040482
5	1400	3615	0.036331
6	1450	1521	0.015286
7	1500	389	0.00391
8	1550	57	0.000573
9	1600	8	0.00008
10	1650	4	0.00004
11	1700	138	0.001387
12	1750	2269	0.022804
13	1800	12 910	0.129747
14	1850	27 677	0.278158
15	1900	25 663	0.257917
16	1950	11 029	0.110843
17	2000	2606	0.026191
18	2050	382	0.003839
19	2100	57	0.000573
20	2150	183	0.001839
21	2200	734	0.007377
22	2250	1421	0.014281
23	2300	1326	0.013326
24	2350	745	0.007487
25	2400	237	0.002382
26	2450	59	0.000593
27	2500	15	0.000151
28	2550	5	0.00005
29	2600	1	0.00001

Algorithm 4 Aggregation algorithm

Require: Let $Sols$ be the set of solutions.
Require: Let $BinSize$ be the rounding factor for the remuneration (50 as an example).
Require: Let $AggregatedData$ be the array of records defined as (X_i, $solsCount_i$) where X_i is the bin label and $solsCount_i$ the amount of solutions associated to that bin.
for $solution_i = 1, 2, \ldots, \mathbf{card}(Sols)$ **do**
  Let Opt_i be the value for the objective function at the current solution.
  $bin \leftarrow Opt_i$ mod $BinSize$
  $X_i \leftarrow bin * BinSize$
  if $AggregatedData_{bin}$ is undefined **then**
    $AggregatedData_{bin} \leftarrow (X_i, 1)$
  else
    $currentValue \leftarrow AggregatedData_bin$ value. Let this value be (X_i, Agg_i)
    $AggregatedData_{bin} \leftarrow (X_i, Agg_i + 1)$
  end if
end for

Finally, the aggregation was made by counting the number of solutions that share the same "bin" (understood as the rounded optimal value) and sorting those bins from lowest to highest. Table 1 represents the density function for the expected remuneration.

4. Results

In this section, we will show the results obtained when the algorithm is applied to a specific case.

4.1. Case definition

In this case, we had defined four different policy types as follows:

- Policy type A:

  - Premium distributed as Lognormal with $\mu = 5.3$ and $\sigma = 0.25$.
  - Insured people distributed as a Poisson with $\lambda = 2$.
- Policy type B:
  - Premium distributed as Lognormal with $\mu = 5$ and $\sigma = 0.2$.
  - Insured people distributed as a Poisson with $\lambda = 1$.
- Policy type C:
  - Premium distributed as Lognormal with $\mu = 4.5$ and $\sigma = 0.3$.
  - This insurance contract type does not allow insured people. It always has 0 as value.
- Policy type D:
  - Premium distributed as Lognormal with $\mu = 6.8$ and $\sigma = 0.15$.
  - This insurance contract type does not allow insured people. It always has 0 as value.

We chose the Lognormal and Poisson distributions because they are some of the most frequently mentioned in actuarial studies, as can be seen in the works of Marlin (1984) and Tzougas et al. (2020) (Lognormal) and Bermúdez and Karlis (2011) and Boucher and Denuit (2006) (Poisson). Regarding the remuneration rules, let us propose the following:

- Commissions: The agent will earn a different percentage for each policy type, so let those percentages be 20% for type A, 30% for type B, 40% for type C and 10% for type D.
- Incentives:
  - Incentive I: In order to be able to earn any incentive, the agent must at least arrange one type A policy and one type B policy.
  - Incentive II: If the total amount of insured people is greater than 10, the agent will earn 500 € as an incentive.
  - Incentive III: If the total amount of premium related to type C policies is greater than 1.200 €, the agent will earn 350 € as an incentive.
  - Incentive IV: If the total amount of policies of types A and B is greater than 12, the agent will earn 400 € as incentive.
  - Incentive V: If the total amount of type A policies is greater than 4, the agent will earn 250 € as an incentive.

Figs. 1 and 2 show the distribution associated for those policy types.

Finally, regarding the internal effort function, and in the sake of exposition clarity, we will make two assumptions: the classifier makes no errors when predicting, and the maximum number of ordered policies in a period is 15 so the last constraint has a limit of 15 units.

4.2. Single instance resolution

Here we will describe how the adapted COBYLA algorithm obtains the optimum value for a given instance. We chose the COBYLA implementation given in the Python NLOPT package (Johnson, 2011).

The list of transactions randomly generated is shown in Table 2 in the format "premium (insured people)" for each policy.

The optimal combination has 2302.76 € as value (reaching incentives I, II & V) and was found in:

- Four type A policies (numerical value: 3.9998843602359764).
- One type B policy (numerical value: 1.183608510036132).
- Zero type C policies (numerical value: 9.982888306925682e−20).
- Almost ten type D policies (numerical value: 9.816507129727892)

Table 2
Transactions randomly generated.

Policy	Type A	Type B	Type C	Type D
1	325,68 (5)	256,84 (3)	116,03 (0)	1179,05 (0)
2	276,62 (4)	189,83 (3)	103,95 (0)	1091,07 (0)
3	246,24 (3)	186,9 (2)	101,71 (0)	1067,1 (0)
4	245,33 (3)	184,86 (2)	96,93 (0)	994,64 (0)
5	242,6 (3)	181,16 (2)	90,3 (0)	979,38 (0)
6	214,97 (2)	154,15 (2)	88,78 (0)	978,81 (0)
7	206,6 (2)	153,62 (1)	84,21 (0)	918,52 (0)
8	201,57 (2)	148,27 (1)	83,38 (0)	905,88 (0)
9	196,03 (1)	143,14 (1)	79,78 (0)	896,61 (0)
10	187,7 (1)	141,22 (1)	79,71 (0)	889,56 (0)
11	184,38 (1)	123,97 (1)	70,73 (0)	886,95 (0)
12	176,06 (1)	123,19 (0)	66,93 (0)	856,69 (0)
13	171,08 (0)	115,62 (0)	52,36 (0)	810,14 (0)
14	165 (0)	113,73 (0)	51,59 (0)	804,42 (0)
15	112,84 (0)	113,35 (0)	50,96 (0)	748,31 (0)

4.3. Complete execution of the algorithm

Finally, we will show the results obtained after a complete execution based on the resolution of 100,000 randomly generated cases. During the batch execution, 499 cases were not resolved by the COBYLA algorithm within 30 s, so they were discarded (the existence of multiple equivalent optimums will be in the next section).

In Fig. 3 the global remuneration for each case is shown. As the figure shows, there were 3 different groups to be analysed (ordered by remuneration average): the smallest, with an average of 1.395 €, the middle, with an average of 1.899 €, and the largest, with an average of 2.302 €.

In Fig. 4, the different distribution for each policy type amounts are shown. It is necessary to highlight the absence of values greater than 0 in type C policies.

Lastly, Fig. 5 shows whether the defined incentive has been obtained for each generated case.

5. Discussion

5.1. Summary of experimental data

The proposed algorithm solves the problem by providing two results: a density function that allows estimating the agent's expected profit and a list of possible sets of transactions with their associated remuneration.

The complete execution showed the following observations related to the incentive compliance level:

- There were 97.277 cases where the amount of type C policies was 0 (97,76%). Only in 2.067 cases (2%), that amount had 1 as value. If the principal wants to influence the agents to arrange more policies of this type, he will have to look for some additional incentives apart from those already defined.
- Incentive I was obtained in 96.331 cases (96,8%).
- Incentive II was obtained in 84.690 cases (85,11%).
- Incentive III was obtained only in 16 cases (0,02%). This fact should make the principal question the effectiveness of it when it comes to influencing the agent's strategy.
- Incentive IV was obtained only in 27 cases (0,03%). As mentioned above, this fact should make the principal question the effectiveness of it when it comes to influencing the agent's strategy.
- Incentive V was obtained in 5.061 cases (5%). This percentage could be misleading in the interpretation of the results, as the highest amount of remuneration (the group with the highest average is covered in the previous section) is only reached when this incentive is achieved.

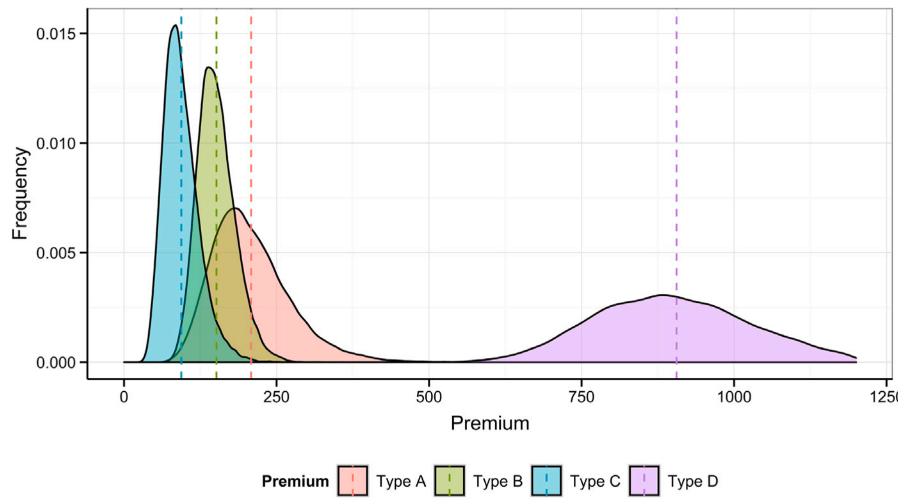

Fig. 1. Premium distribution by policy type.

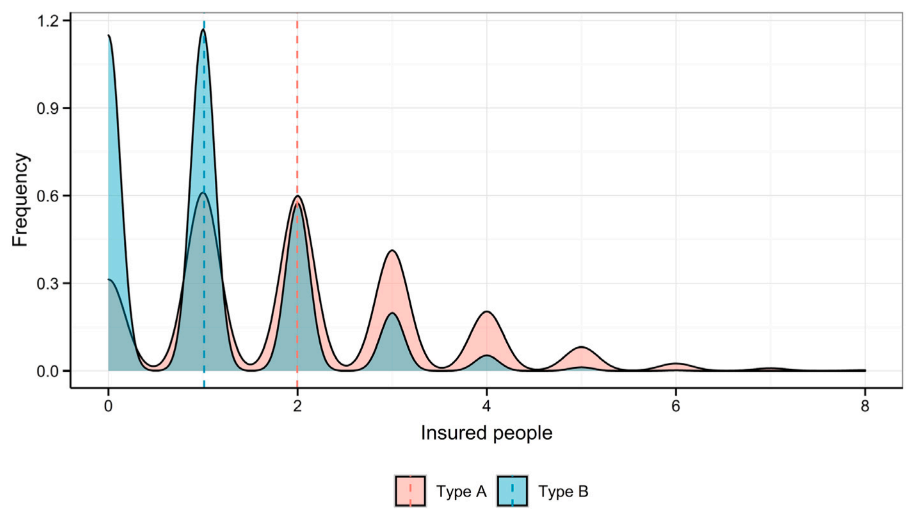

Fig. 2. Insured people distribution by policy type.

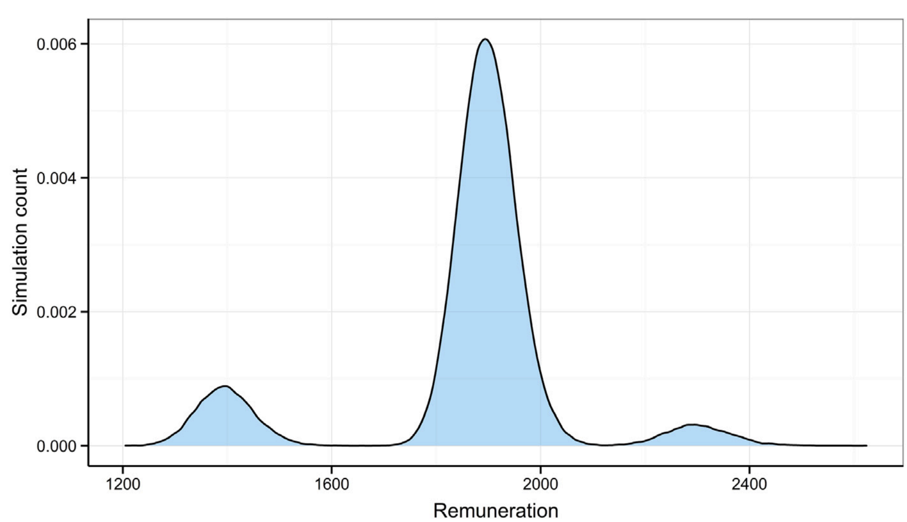

Fig. 3. Remuneration.

The conclusions drawn from the analysis of the number of policies present are as follows:

- Type C policies are not profitable (from the agent's point of view) under the current incentive scheme. This conclusion is

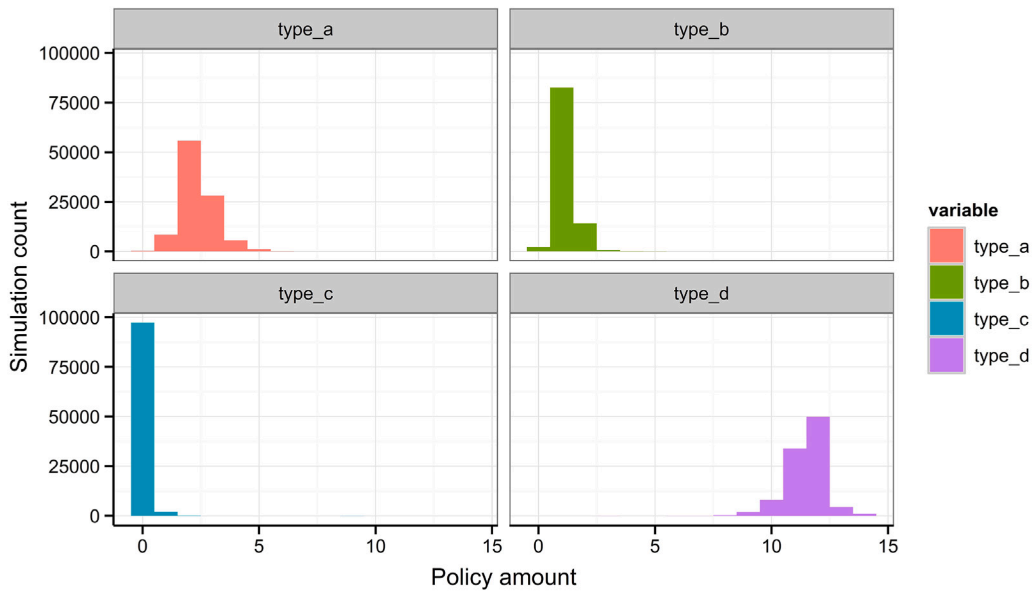

Fig. 4. Policy arrangement by type.

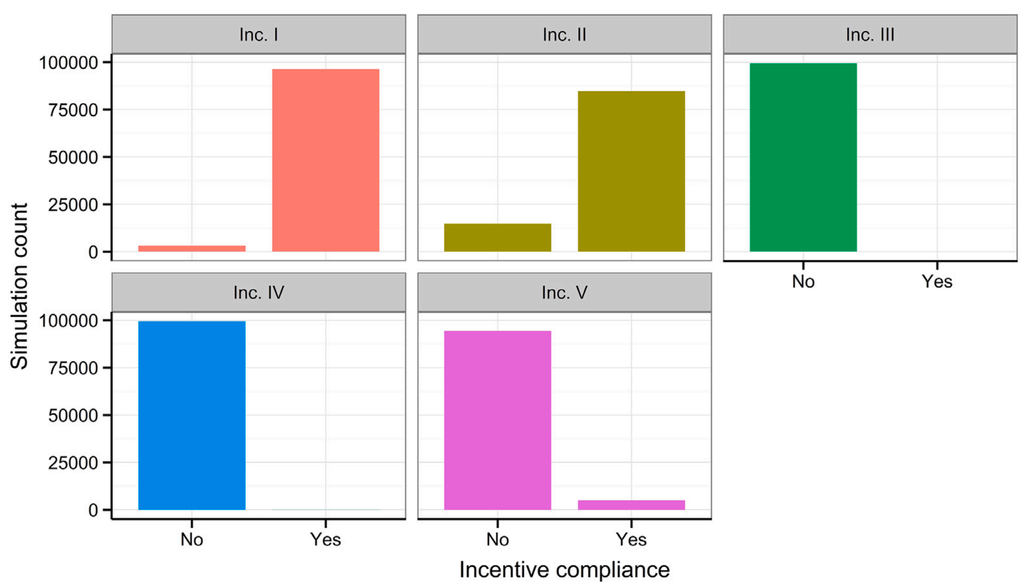

Fig. 5. Incentive achievement levels.

also remarked in the prior analysis (0,02% compliance level for incentive III)

- Regarding the first remuneration group (with smallest amount, an average of 1.395 €) the average amounts for policies were: 1827 type A polices, 1 type B policy, 0 type C policies and 12,155 type D policies. Most of the scenarios belonging to this group only succeeded in achieving the first incentive.
- Using the same analysis on the second group (with intermediate amount, an average of 1.899 €), the results showed that the average amounts for policies were the following: 2336 type A policies, 1174 type B policies, 0 type C policies, and 11,4624 type D policies. Most scenarios belonging to this group only succeeded in achieving both incentives I and II.
- The analysis made on the last group (the one with the greatest amount, an average of 2.302 €) showed that the average amounts for policies were the following: 4322 type A policies, 1263 type B policies, 0 type C policies, and 9408 type D policies. Most scenarios belonging to this group only succeeded in achieving incentives I, II and V.

As a result of the prior insights, the principal would be able to identify topics regarding the incentive scheme under study like the following. Firstly, the incentive III (specifically designed to encourage the arrangement of type C policies) is not working as expected, and neither is incentive IV. Secondly, the average of 1 type B policy suggests that this policy is instrumental: it is necessary to achieve incentive I, but once it has been is achieved, type B policies are not useful for the agent for maximizing its profit. Thirdly, the fact that the highest amounts of remuneration are reached with higher amounts of contract type A policies (4322 versus 1827) and lower amounts of contract type D policies (9408 versus 12,155) should be highlighted.

5.2. Known challenges for the proposed method

First, depending on the definition of the policy types, the existence of a single global optimum cannot be guaranteed while solving the associated optimization problem. This is a common event in solving combinatorial optimization problems, but for this problem, it could happen more frequently, as two different types of policy may have the same premium value and number of insured persons. Given these circumstances, if one of them was a part of the optimal solution, the other could also be part of another equivalent optimum solution, and both would have the same objective function value. This could be possible if their exchange did not affect the calculation of the incentives.

Secondly, it is very common to use linear functions or piece-wise linear functions to define the remuneration concepts. This method can deal with non-linear functions, as it uses a non-linear solver. However, the use of this kind of function could lead to non-feasible problems. Furthermore, the use of complex mathematical functions could lead to a misunderstanding on the agent's part, even when understanding the contract (with the principal), since advanced training in mathematics cannot be assumed as part of the agent's skills.

Third, given that the agent knows the incentives before carrying out his work, the proposed method cannot anticipate the way in which said knowledge modifies the agent's ex ante behaviour. This is especially relevant when training the classification algorithm, which is necessary when adjusting the theoretical solutions to the real behaviour, since the principal only perceives the behaviour ex post.

Finally, there may be situations in which the agent increases or decreases the amount of effective work that can be done within a period of time. The proposed method should be robust enough for this type of situation when these variations have already occurred and there is enough historical data for the supervised learning algorithm to accurately identify these behaviour patterns.

6. Conclusions

In this paper we have proposed a method that allows the principal to estimate the remuneration that an agent will receive under the conditions established in the contract between them. However, the proposed method is completely reactive. That means, the principal chooses the new conditions of the contract and evaluates the impact that these modifications have on the agent's remuneration. An unresolved aspect, which could be studied as future work, is to study the inverse problem consisting of determining how to alter the terms of the contract so that it guides the arrangement of certain policy types to be made by the agent, prioritizing them over others.

CRediT authorship contribution statement

Fernando Turrado García: Conceptualization, Formal analysis, Investigation, Methodology, Project administration, Software, Validation, Writing – original draft. **Ana Lucila Sandoval Orozco:** Conceptualization, Formal analysis, Investigation, Methodology, Project administration, Software, Validation, Writing – original draft. **M. Pilar García Pineda:** Conceptualization, Formal analysis, Investigation, Methodology, Project administration, Software, Validation, Writing – review & editing. **Luis Javier García Villalba:** Conceptualization, Formal analysis, Investigation, Methodology, Project administration, Software, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgements

This research work was supported by Ocaso, S.A. Insurance and Reinsurance Company under the agreement made with Universidad Complutense de Madrid for the execution of the industrial research project developed by the company.

References

Albrecht, S. V., & Stone, P. (2018). Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, *258*, 66–95.

Bermúdez, L., & Karlis, D. (2011). Bayesian multivariate Poisson models for insurance ratemaking. *Insurance: Mathematics & Economics*, *48*(2), 226–236.

Boucher, J.-P., & Denuit, M. (2006). Fixed versus random effects in Poisson regression models for claim counts: A case study with motor insurance. *Astin Bulletin*, *36*(1), 285–301.

Castiglioni, M., Marchesi, A., & Gatti, N. (2022). Bayesian agency: Linear versus tractable contracts. *Artificial Intelligence*, *307*, Article 103684.

Chen, K., Wang, X., Huang, M., & Ching, W.-K. (2017). Salesforce contract design, joint pricing and production planning with asymmetric overconfidence sales agent. *Journal of Industrial and Management Optimization*, *13*(2), 873–899.

Chod, J., & Lyandres, E. (2021). A theory of ICOs: Diversification, agency, and information asymmetry. *Management Science*, *67*(10), 5969–5989.

Coulom, R. (2009). The monte-carlo revolution in go. In *The Japanese-French frontiers of science symposium (JFFoS 2008), Vol. 115* (pp. 192–201).

Dantzig, G. B. (1957). Discrete-variable extremum problems. *Operations Research*, *5*(2), 266–288.

Dudzik, W., Nalepa, J., & Kawulok, M. (2021). Evolving data-adaptive support vector machines for binary classification. *Knowledge-Based Systems*, *227*, Article 107221.

Henig, M. I. (1990). Risk criteria in a stochastic knapsack problem. *Operations Research*, *38*(5), 820–825.

Holmstrom, B., & Milgrom, P. (1987). Aggregation and linearity in the provision of intertemporal incentives. *Econometrica*, *55*(2), 303–328.

Huang, B., Zhang, X., & Bi, Q. (2022). The nonlinear effect of shareholder ownership structure on a firm's cash holdings: Type I and Type II agency problem perspectives in China's split-share reform. *International Review of Economics & Finance*, *77*, 493–504.

Johnson, S. G. (2011). The NLopt nonlinear-optimization package.

Lai, C., Liu, L., & Li, R. (2021). The optimal solution to a principal-agent problem with unknown agent ability. *Journal of Industrial and Management Optimization*, *17*(5), 2579–2605.

Lee, I. G., Zhang, Q., Yoon, S. W., & Won, D. (2020). A mixed integer linear programming support vector machine for cost-effective feature selection. *Knowledge-Based Systems*, *203*, Article 106145.

Luo, Q., & Saigal, R. (2021). Dynamic multiagent incentive contracts: Existence, uniqueness, and implementation. *Mathematics*, *9*(1).

Ma, J., Yang, L., & Sun, Q. (2021). Adaptive robust learning framework for twin support vector machine classification. *Knowledge-Based Systems*, *211*, Article 106536.

Marlin, P. (1984). Fitting the log-normal distribution to loss data subject to multiple deductibles. *The Journal of Risk and Insurance*, *51*(4), 687–701.

Morton, D., & Wood, R. (1998). On a stochastic knapsack problem and generalizations. *Advances in Computational and Stochastic Optimization, Logic Programming and Heuristic Search*, *1*(1), 149–168.

Powell, M. J. D. (1994). A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in optimization and numerical analysis* (pp. 51–67). Dordrecht: Springer Netherlands.

Powell, M. J. D. (1998). Direct search algorithms for optimization calculations. *Acta Numerica*, *7*, 287–336.

Prat, J., & Jovanovic, B. (2014). Dynamic contracts when the agent's quality is unknown. *Theoretical Economics*, *9*(3), 865–914.

Schättler, H., & Sung, J. (1993). The first-order approach to the continuous-time principal–agent problem with exponential utility. *Journal of Economic Theory*, *61*(2), 331–371.

Shao, T., Zhang, H., Cheng, K., Zhang, K., & Bie, L. (2021). The hierarchical task network planning method based on Monte Carlo Tree Search. *Knowledge-Based Systems*, *225*, Article 107067.

Slyke, R. V., & Young, Y. (2000). Finite horizon stochastic knapsacks with applications to yield management. *Operations Research*, *48*(1), 155–172.

Turnbull, G. K., Waller, B. D., & Wentland, S. A. (2022). Mitigating agency costs in the housing market. *Real Estate Economics*, *50*(3), 829–861.

Turrado García, F., García Villalba, L. J., & Ana Lucila Sandoval Orozco, T.-H. K. (2019). A comparison of learning methods over raw data: forecasting cab services market share in New York City. *Multimedia Tools and Applications*, *78*(21), 29783–29804.

Turrado García, F., García Villalba, L. J., & Portela, J. (2012). Intelligent system for time series classification using support vector machines applied to supply-chain. *Expert Systems with Applications*, *39*(12), 10590–10599.

Tzougas, G., Yik, W. H., & Mustaqeem, M. W. (2020). Insurance ratemaking using the Exponential-Lognormal regression model. *Annals of Actuarial Science*, *14*(1), 42–71.

Williams, N. (2015). A solvable continuous time dynamic principal–agent model. *Journal of Economic Theory*, *159*, 989–1015, Symposium Issue on Dynamic Contracts and Mechanism Design.