Reliably Assessing the Effectiveness of a Plan Using Models of Varying Fidelity and Under Time Constraints

Steven de Jong, Wouter Noordkamp, Nick van der Poel, Selmar Smit
TNO Defence, Safety & Security, The Netherlands
Steven.deJong@tno.nl, Wouter.Noordkamp@tno.nl, Nick.vanderPoel@tno.nl, Selmar.Smit@tno.nl

ABSTRACT

Assessing the effectiveness of a plan, given multiple potential scenarios, is a common problem for analysts, especially in the military domain. This problem can seriously impact the safety of the people that are involved in planned missions. More precisely, the availability of multiple models, with varying levels of fidelity, leads to the complex task of selecting the best model(s) to assess the effectiveness of a plan. Under time constraints, optimal model selection depends not only on the fidelity of the models at hand, but also on the nature of the possible scenarios the plan applies to, such as the potential presence of stochastic variables and the number of different scenarios that have to be evaluated in order to obtain a reliable estimate of the true effectiveness of the plan. In this paper, two algorithms are presented to maximize the reliability of the obtained plan effectiveness under time constraints. To this end, the algorithms select the best model(s) as well as the most appropriate scenarios. Both algorithms have been tested on synthetic data as well as on two Navy-related use cases. Results show that both algorithms reach a higher level of reliability within the given amount of time than conventional approaches. Thus, they allow analysts to better assess the effectiveness of their plans and therefore they increase the safety of everyone involved in planned missions.

ABOUT THE AUTHORS

Steven de Jong obtained a PhD in artificial intelligence in 2009. He subsequently performed postdoctoral research at Maastricht University (The Netherlands) and Brussels University (VUB, Belgium) on topics such as multi-agent systems, computational fairness and swarm intelligence. He currently works at TNO, the Netherlands organization for applied scientific research, as a technical consultant in the modelling and simulation domain. In his work, he focusses on bridging the gap between IT support systems and system operators.

Wouter Noordkamp graduated in Applied Mathematics at the University of Twente in Enschede, the Netherlands, in 1999. Since then he has been working at TNO in the Operational Analysis department. His main activities are building quantitative models and using them in support of the Royal Netherlands Navy in the fields of acquiring new materiel and improving the deployment of new systems. He started in Anti-Air Warfare, but currently works on the deployment of sensors and systems in Anti-Submarine Warfare.

Nick van der Poel graduated in Astronomy at the University of Utrecht, the Netherlands, in 2010. After 2 years of working as a software engineer, he started working at TNO in the department of Modelling, Simulation & Gaming. Here, he is responsible for the development and maintenance of an extensive simulation suite that is used for answering air and missile defense related questions.

Selmar Smit did his PhD at the Vrije Universiteit Amsterdam, the Netherlands, on the topic of parameter tuning of metaheuristics, model selection and response surface modelling. He currently works as a data scientist at TNO, where his main focus is on predicting human behavior.

Reliably Assessing the Effectiveness of a Plan Using Models of Varying Fidelity and Under Time Constraints

Steven de Jong, Wouter Noordkamp, Nick van der Poel, Selmar Smit
TNO Defence, Safety & Security, The Netherlands
Steven.deJong@tno.nl, Wouter.Noordkamp@tno.nl, Nick.vanderPoel@tno.nl, Selmar.Smit@tno.nl

INTRODUCTION

In many settings, there is a need to measure the effectiveness of a plan. This need occurs especially (but not exclusively) before and during military missions. For example, a plan may be needed to position a number of units such that their capability for detecting incoming threats is maximized. The direction the threat will come from is not known beforehand, and therefore we need to determine the effectiveness of the plan to defend our assets for an arbitrarily large number of possible directions.

More generally speaking, a plan often has to be evaluated against a multitude of scenarios, i.e. a set of parameters describing the circumstances the plan is going to be executed in, such as different locations or different environmental conditions. Moreover, variables represented in scenarios may be stochastic, for example the direction from which a threat arises, or the probability of hitting an intended target. This means that the same scenario should be evaluated multiple times to obtain a reliable estimate of the true effectiveness of a plan in that scenario.

The effectiveness of a plan is generally assessed by performing computer simulations, in which models of real-world phenomena are used. Some of these models, such as sonar or radar models that describe how well units are capable of detecting threats, may have a very high fidelity and may require a long calculation time. Under time constraints, e.g. due to unforeseen circumstances during a mission, the analyst thus faces the challenge of not being able to accurately determine the plan's effectiveness for every potential scenario. Similarly, only a limited number of repetitions can be performed for scenarios that contain stochastic variables, leading to a potential erroneous estimation. This is a high-impact problem, for example because a plan is assessed as more effective than it is in reality. Unexpected complications and risks to units' safety may occur once this plan is used in practice.

To tackle this problem, faster models with a lower fidelity are often developed in addition to the aforementioned high-fidelity models. For example, the complex calculations required by a sonar model can be performed ahead of time on a number of relevant scenarios and stored in a lookup table. This table can then be used to obtain outcomes for scenarios similar to those stored. Obviously, the table offers a much faster model than the original sonar model, but it is also much less accurate, especially when evaluated scenarios differ from those stored in the table.

Although the availability of faster, lower fidelity models offers a potential solution to the problem of time constraints, it also complicates the analyst's task to perform the assessment of plan effectiveness. Selecting which scenarios to evaluate, with which model, and how often to repeat the evaluation of a single scenario in case the scenario has stochastic variables, is not a trivial task. Different choices can lead to very different outcomes. We have observed that the consequences of this selection may be easily underestimated. In some assessments, the unknowns within a scenario lead to a large number of stochastic variables with a high impact on observed outcomes. In this case, it may be a good choice to use a faster, lower fidelity model to evaluate as many repetitions of the scenario as possible, even though each individual evaluation is known to be relatively inaccurate. In other assessments, the focus should be on the accuracy of each individual evaluation. For example, if an approximate model overestimates the probability of detecting a threat in a given direction, this could have disastrous results.

In this paper, we propose two algorithms that can be used to find a scheme of evaluations that optimizes the selection of scenarios, models and repetitions. Within the next section, the complexity of the problem at hand is discussed in more detail. Then, we address the work that has already been carried out on this topic and discuss two Naval use cases. This is followed by a description of the two developed algorithms, along with their performance as compared to current approaches. Finally, we summarize our work and discuss implications for operational practice as well as future research.

COMPLEXITY OF MODEL SELECTION FOR MULTIPLE EVALUATIONS

As outlined above, this paper is concerned with the complex issue of evaluating the effectiveness of a plan given multiple potential scenarios, the potential presence of stochastic variables in these scenarios, and multiple candidate models to perform the evaluation with a limited time budget.

More generally, the work presented here applies to optimization problems that require multiple evaluations of one or more instances[1], in the presence of multiple models of different fidelity and complexity that can be used to evaluate any desired measure related to these instances. The work is especially relevant if the time to perform evaluations is not sufficient to thoroughly evaluate every instance with the most reliable model.

There are two factors that determine whether an optimization problem requires multiple evaluations. First, the optimization problem may contain stochastic variables.[2] A single evaluation of a given instance is in this case no more than one sample from a distribution of possible outcomes, and as such, it is a highly unreliable estimate of the true outcome for an instance. Second, the optimization problem may require evaluating multiple instances due to the nature of the problem (e.g. when there are different plan variants that the analyst has to consider), but there may be insufficient time to evaluate all instances with a high reliability. These factors can be combined to create four situations, which have been summarized for clarity in Table 1 below.

Table 1: Overview of the various factors that can cause the need for multiple evaluations. Highlighted in green are the combinations that are elaborately discussed in this paper.

Nature of the problem at hand	Stochastic variables present	Only deterministic variables
Single instance	x	
Multiple instances required		x

In Table 1, for one of the four situations a best approach is trivial, i.e. in the top right of the table. In this situation, the problem involves only deterministic variables and only one instance. Since the instance is deterministic, a single evaluation of this instance suffices, and therefore, only the expected calculation times of the models available to the analyst influence the choice for a model. Therefore, the most reliable model (with a calculation time less than the time available) is the best choice.[3]

Looking at the other three situations, we see that finding a best approach is non-trivial. For the **top left** corner of Table 1, we can find some work in the literature on determining the reliability of a given model, given a number of evaluations for a given instance. However, this is not the case for the situation in the **bottom right** corner of Table 1, as there is little work devoted to selecting which subset of instances to evaluate to the get the most reliable result within a limited amount of time. Finally, the situation in the **bottom left** of Table 1 deals with the case that the analyst needs to determine which instances to test, with which model, and with how many evaluations for each instance. In this paper, we introduce two approaches for each of the two other non-trivial situations (i.e. in the top left and in the bottom right of Table 1) and argue that they can be combined in order to approach this situation as well.

RELATED WORK

In the last decade, a considerable amount of work has been dedicated to predicting the outcome of an algorithm or a model (Hutter, 2012). Starting in the 1970's, *algorithm selection* was identified as the problem of selecting the best algorithms or models for a certain instance (Rice, 1976; Roberts, 2007; Kotthoff, 2012). There are many methods to perform algorithm selection, such as statistically (Gagliolo, 2006), with regression (Fink, 1998) or with neural networks (Smith-Miles, 2009). Our approach uses a Gaussian Kriging algorithm (Chiles, 1999); however any of the alternative approaches would lead to comparable results (Hutter, 2012). For an exhaustive overview of prediction methods, including Gaussian Kriging, as used in the paper at hand, see Hutter (2012).

[1] Where we previously used the more intuitive term *scenario*, we use the more accurate term *instance* in the remainder of this paper. Formally, an instance is a combination of (model) parameters, such as environmental characteristics, used to perform a calculation.

[2] We note that stochasticity may arise as a result of stochasticity in the instance (e.g. some parameters may relate to random distributions) or as a result of evaluating the instance with a stochastic model.

[3] The issue of selecting the most reliable model for evaluating a single instance, or in fact all instances available, is commonly referred to as *algorithm selection* (see next section). We note that this is different from what we discuss under 'multiple instances required', because in our case, a selection has to be made as to which **subset** of instances are evaluated, since there is no time to evaluate all of them with the most reliable model.

Howe et al (2000) used linear regression to predict how both a planner's runtime and its probability of success depend on various features of the planning problem (this is called *response surface modelling*). Similar to our approach, they applied these predictions to decide which algorithm, from a given set of algorithms, should be run in order to optimize a performance objective. Specifically, they used the expected accuracy (reliability) of an algorithm, divided by the expected runtime of this algorithm, as the performance objective.

Other research was aimed at automatically identifying the features of optimization problems that have a strong influence on an algorithm's outcomes. The literature on the topic of search space analysis for example, has proposed a variety of features correlated with the difficulty of the problem at hand. Prominent examples include fitness distance correlation (Jones, 1995) and autocorrelation length (ACL) (Weinberger, 1990). As problem difficulty is obviously related to both runtime and reliability, such work is useful to automatically determine the features that can be relevant for algorithm selection. In this paper, in order to focus on the instance-selection methods, we have chosen to use a handmade set of features, devised by subject matter experts, rather than to automatically determine them.

We define *instance selection* as the act of selecting which instances to evaluate along with determining how many repetitions need to be performed. After selecting these instances, algorithm selection is performed to determine the model that is used for evaluating the selected instances. Logically, the selected instances are those that have the best expected contribution to the reduction of uncertainty in the estimated outcome. In that sense, instance selection is very similar to the selection mechanism applied in parameter-tuning methods with response-surface modelling such as Sequential Parameter Optimization (Bartz-Beielstein, 2005). However, the instance with the highest expected value is selected in this literature, rather than the instance with the highest expected reduction in uncertainty, as we do in this paper. As with our approach, the authors use a Kriging model to establish expected values.

In summary, the approach presented in this paper uses ideas behind algorithm selection, using prediction models from the field of response surface modeling. It extends these approaches with instance selection strategies.

USE CASES

To illustrate the complexity of model and instance selection and clarify and test the workings of our algorithms, we use two use cases from a Naval domain throughout this paper. Both use cases are based on concrete questions we have received from the Dutch Navy; the second use case was brought forward by the Navy when we presented the results concerning the first use case.

The first use case relates to the problem setting of a single instance with a stochastic variable (c.f. upper left in Table 1). Here, the quality of a Naval task group configuration has to be evaluated against a threat (e.g. a submarine). The direction from which the threat arises is unknown. Therefore, evaluations have to be performed for many potential directions. In this case, the instance is the same for every evaluation (i.e. the task group configuration of the defending units is the same every time), but the direction of the incoming threat is unknown, requiring it to be modelled as a stochastic variable.

The second use case relates to the problem setting where we need to evaluate multiple instances with only deterministic variables (c.f. lower right in Table 1). Here, a ship has to determine the safest route through a region with potential mobile underwater threats (e.g. submarines). In such a scenario the characteristics of the seabed have a strong influence on the performance of the ship's sonar, and therefore on the detectability of an underwater threat. Some routes may take the ship through areas with severely reduced detectability. Ideally, the detectability of threats should be assessed for each location on each potential route, as well as in each direction from that location, so that the safest route may be chosen. Obviously, even when the characteristics of the seabed are fully known (i.e. there is no stochasticity in the description of the sea floor), this results in far too many calculations to be feasible in limited time. Therefore, in order to ensure a safe passage, the detection capabilities of the ship have to be derived as reliably as possible, for as many locations in the region as possible, in the given limited time.

APPROACH

In this section, we discuss the algorithms we devised for the two problem settings under consideration (i.e., a single instance with stochastic variables, and multiple deterministic instances). Before discussing the approach for these settings separately, we first discuss the common elements below.

Common Elements

The first issue to be addressed when choosing a model from a set of models is the required ability to express the accuracy of these models when compared to reality.[4] In practice, even the most accurate model differs from reality. In our work, we assume this difference is not present. We simply cannot do better than the best model of reality. Therefore, the best known model is termed the **reference model**.

Given the reference model and a set of approximate models, we create a database of paired example outcomes. With a large number of instances, we perform a number of evaluations with each model, with a (constant) number of repetitions if the instances are stochastic; all results for all repetitions are saved. With more instances and/or more repetitions in the database, we will be able to obtain a better estimate of each model's reliability in comparison with the reference model outcome. In the database, we do not only save the model outcomes, but also the time required to obtain each outcome (i.e. the calculation time).

A Single Instance With Stochastic Variables

The first use case illustrates a situation where the instance evaluated has one stochastic variable: we need to detect a threat from an unknown direction before it has the chance to fire. In this case, with a stochastic variable (or multiple stochastic variables), we find uncertainty with the outcome due to a limited number of instance evaluations. This uncertainty is present for each model, including the reference model, and can be reduced by performing additional evaluations. With no time limit, we would be able to evaluate every possible stochastic variation and obtain an uncertainty of zero on the obtained outcome. In practice, we are only able to evaluate a subset of all possible variations, and thus have a remaining level of uncertainty.

When outcomes of the approximate models are compared to outcomes of the reference model, we find a second source of uncertainty. Generally, the difference (**bias**) between the outcomes of a reference model and those of an approximate model is not constant, but distributed stochastically, depending on the instance at hand and on random factors in the models at hand. When we correct or tune the approximate model outcome by adding or subtracting the mean bias (which is the best estimate of the true outcome given the approximate outcome), we introduce uncertainty concerning the model bias. In contrast to the uncertainty of a limited number of evaluations, the model bias uncertainty cannot be reduced by performing more evaluations.

The algorithm proposed in this section requires a single formula that can express the overall uncertainty of an outcome, given the two uncertainties concerning a limited number of evaluations and concerning the model bias and a certain calculation time. The formula is straightforward. Given that

- σ_T denotes the total uncertainty (expressed as standard deviation);
- σ_B denotes the uncertainty concerning approximate model bias; and
- σ_L denotes the uncertainty on the outcome due to a limited number of evaluations,

we obtain the formula $\sigma_T = \sqrt{\sigma_B^2 + \frac{\sigma_L^2}{n}}$.

For the reference model, σ_B is always 0 since it does not have a bias. For an approximate model, we calculate the model bias for every individual instance. The uncertainty concerning approximate model bias is calculated by determining the variance of the biases for all instances.

[4] We note that some models may yield more than one outcome, resulting in more than one estimate of accuracy or reliability. (We use these terms interchangeably, which is somewhat incorrect, but helps the story of the paper.) Also, reliability may depend on the desired application. In the remainder of this paper, we work with models that have a single outcome and assume reliability is determined for the application at hand.

The parameter *n* describes the number of times the model in question can be used within a certain available amount of calculation time. We note that the calculation time of a model is, in general, not constant, but taken from an unknown distribution. Therefore *n* needs to be established empirically.

Given a large example database containing outcomes and calculation times for a number of models, we can use the formula above to estimate the effects of the two sources of uncertainty. With limited total calculation time, the formula predicts that fast, low fidelity models may have a lower total uncertainty than slow, high fidelity models. The value of *n* is larger for faster models, which can compensate for the large σ_B^2 they may have. On the other hand, with increasing calculation time, high fidelity models are also able to perform a sufficient number of evaluations to decrease the uncertainty due to a limited number of evaluations.

Multiple Deterministic Instances

Our approach for multiple deterministic instances relies on one central idea, which is that we can predict outcomes for instances that have not been evaluated yet and iteratively improve our prediction over time.

In order to predict outcomes, we use the aforementioned Gaussian Kriging algorithm (Chiles, 1999). Using this algorithm, outcomes can be predicted for unknown instances, given a number of example instances and related outcomes that were obtained ahead of time. The algorithm uses known similarities between instances to perform this prediction. After all, it is quite probable that instances with similar characteristics yield a similar outcome. In Figure 1, we provide a schematic overview of the approach, centered around Gaussian Kriging.

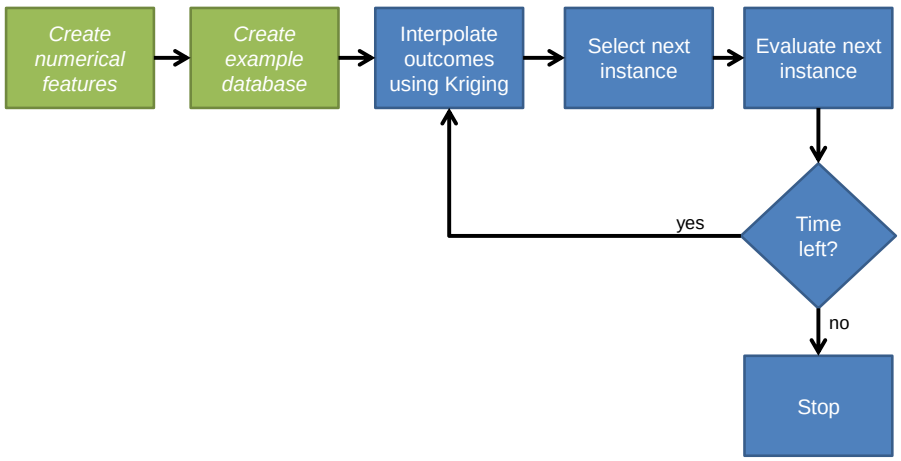

Figure 1: Obtaining the highest reliability at any time when multiple deterministic instances need to be evaluated.

The approach starts with two steps that can be performed ahead of time (in green and in italic font). First, in order to determine the similarity between instances, a limited number of **numerical features** have to be established that adequately describe the characteristics of instances that most influence outcomes. For example, in our Naval use case in which a ship is detecting underwater threats during a route, we asked subject matter experts which are the characteristics of an open sea environment that most influence a sonar model's ability to detect underwater threats in a timely manner. Characteristics such as the profile of the seabed (slope, roughness) and the water temperature turned out to be of great influence. For those characteristics that are not directly represented as a number (or at most a few numbers), a mapping needs to be defined that yields numerical features. For example, the roughness of the seabed in the direction of the sonar beam may be represented by the variance of the gradient over a distance of several kilometers in that direction. The second step we can perform ahead of time is the creation of an **example database**, as outlined at the beginning of this section. In addition to outcomes and calculation times for each instance and model at hand, we also need to store all relevant features of each individual instance.

The following steps in the approach (in blue and regular font) are performed whenever we want to obtain reliable outcomes in the shortest possible time. These steps are performed iteratively. In other words, these are steps that are

performed online. First, all currently known information is used by the Kriging algorithm to **obtain expected outcomes** for instances that have not yet been evaluated. This process is known as interpolation. Initially, the known information is limited to what we have stored in the example database ahead of time. Over time, after a number of iterations, outcomes that are obtained in previous iterations are also used by the Kriging algorithm to gradually improve the reliability of the expected outcomes. One of the great advantages of the Kriging algorithm is that it not only provides an expected outcome, but also gives a confidence interval on this outcome. When the interpolation is of sufficient quality, e.g. given a dense and relevant example database, confidence on the interpolated outcome can be very high. Therefore, an analyst will be able to use an interpolated outcome as if it were an outcome obtained by actually performing an evaluation with one of the available models.

The second online (blue box with regular font) step found in Figure 1 concerns the **selection of the next evaluation** to perform with a certain available instance and model. At any given time, we may have (1) a number of instances that have only an interpolated outcome, (2) a number of instances that have been evaluated by the reference model, and (3) a number of instances that have been evaluated by one of the approximate models. The question is which instance to select and with which model to evaluate it with, to ensure that the reliability of the total outcome increases as quickly as possible.[5] One strategy to perform this task is to select the combination of instance and model that most reduces the total uncertainty over time (e.g., reducing the uncertainty from, say, 50 to 40 in one second is better than reducing it from 50 to 30 in five seconds). In order to be able to use this strategy, we need to be able to predict the uncertainty on a model outcome for a specific instance. Fortunately, the Kriging algorithm and the example database provide us with the tools to do so. Intuitively, we can use Kriging to predict the uncertainty for an unknown instance by interpolating the known uncertainties for known instances in the example database. Similarly, we can use Kriging to predict how much time a model would need to evaluate the unknown instance. With these predictions, we can easily calculate the expected reduction in uncertainty over time given the current uncertainties. Thus, the instance and model are selected that are expected to reduce total uncertainty over time in the best manner.

The selected instance and model are then used, in the third online (blue box with regular font) step found in Figure 1, to perform an **instance evaluation**. If there is still time left or the analyst does not stop the iterative process, the obtained result is added to the known information and the Kriging algorithm is executed again on the updated known information in order to obtain more reliable expected outcomes. Should the analyst stop the iterative process, we are certain that we are providing the most reliable total outcome possible at that time. In our use case, the ability for the ship to detect threats will have been assessed for each location and direction that the analyst indicated as important. This will produce the most reliable assessment that is possible in the given time. Moreover, the analyst has a clear indication of the reliability, which helps in determining the risks of accepting the obtained assessment, which is more or less indicative of the true ability to detect threats.

The instance selection strategy outlined in Figure 1 and explained above, i.e. online optimization, is appropriate for a problem setting where available calculation time is limited but unknown. The strategy tries to decrease uncertainty as quickly as possible and provide the best result at any time. However, there is a disadvantage to the online optimization strategy, namely that it will regularly evaluate the same instances multiple times. The estimate provided by Kriging often has a high uncertainty. A fast approximate model may be able to best reduce this uncertainty quickly, but it will not remove the uncertainty completely. Therefore, at a later moment, if there is time, the strategy will select the instance in question again and evaluate it with a more reliable model to further reduce uncertainty. To address this issue when the total calculation time is in fact known, we developed a second instance selection strategy, i.e. an offline optimization strategy. Before the start of evaluations, a genetic optimization method is used that assigns precisely one model (or no model at all) to each instance (i.e. it creates an evaluation schema). The fitness function performs sequential Kriging to predict outcomes and confidence intervals for each evaluation. It thus returns the expected final confidence on the overall outcome. Then evaluations are performed according to the schema dictated by the genetic optimization method. Although this strategy saves valuable calculation time, we run the risk that it cannot accurately estimate the entire outcome beforehand and therefore could propose the wrong schema. For the offline optimization strategy, the size of the initial outcome database is very important, because a well-sized database ensures that the estimates done in the optimization method are of a sufficient quality to provide us with a

[5] We note that the task performed by this strategy is also performed by analysts in the field at the moment. A very basic strategy that is often used currently is to repeatedly select a random untested instance, or an instance the analyst thinks is important, and to evaluate it with the reference model. This strategy does not use any of the information acquired by the previous components, and therefore may perform very poorly.

good evaluation schema. With a small database, the estimates may be rather faulty, and therefore, the optimization method yields a schema that in fact does not lead to the best result possible within the given calculation time.

RESULTS

Given the sensitivity and the complexity of the presented use cases, we have chosen a controllable and unclassified test set, namely synthetic test problems, to generate our results.

We use two famous continuous optimization problems that are known to be hard to optimize but do have a general structure. This makes sure that it is possible to use the similarity between two instances for reasoning however, also makes sure that it is not possible (or at least very hard) to precisely interpolate. More precisely, the Griewank function is utilized as the output function. In order to simulate models with different reliabilities, for every model with index *m* and total number of features *n* only the first *n* - *m* features are used. All other features are set to 0. The more features are set to 0, the less reliable the model. Therefore we end up with a set of models with decreasing reliabilities. For the simulation of model runtime, we have used the Schwefel function, in which the outcome is divided by m^2 for the model with index *m*. Thus, a model with a higher index is both less reliable as well as faster, which adequately matches our needs for models of varying complexity.

A Single Instance With Stochastic Variables

One of the nice advantages of the proposed algorithm is that it intuitively shows how the reliability of each model changes when the time budget is increased. In Figure 2, the uncertainty on the outcome for each of three models (1-3, with 1 as the reference model) is depicted for a time budget of 600s, 1200s and 4800s. It is clear that in the first case (budget of 600s), Model 1 has the highest total uncertainty, and Model 3 has the lowest uncertainty. However, by increasing the budget to 4800s, Model 1 becomes the best model, since the uncertainty concerning the bias for the other models does not decrease over time. Model 1 does not have this uncertainty since it is the reference model.

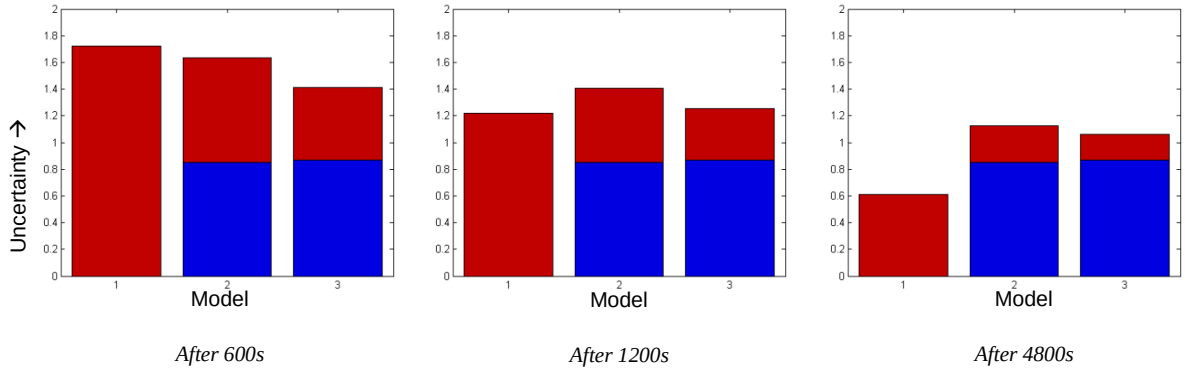

Figure 2: A decreasing uncertainty over time for three models of varying complexity. We show three distinct time budgets. In the graphs, the red area represents the uncertainty due to limited calculation time, and the blue area represents the uncertainty concerning the model bias.

Figure 3 shows the total uncertainty given any time budget instead of for three distinct time budgets. From this figure, we can conclude that Model 2 is never the best model. For a low time budget (up to 1,000s), Model 3 provides a lower total uncertainty on the obtained outcome than Model 2. For a higher calculation budget, the reference model (Model 1) outperforms Model 2. This fact shows that our analysis cannot only be used to determine which model to use given a certain time budget, but also to determine whether a certain model has any added value at all.

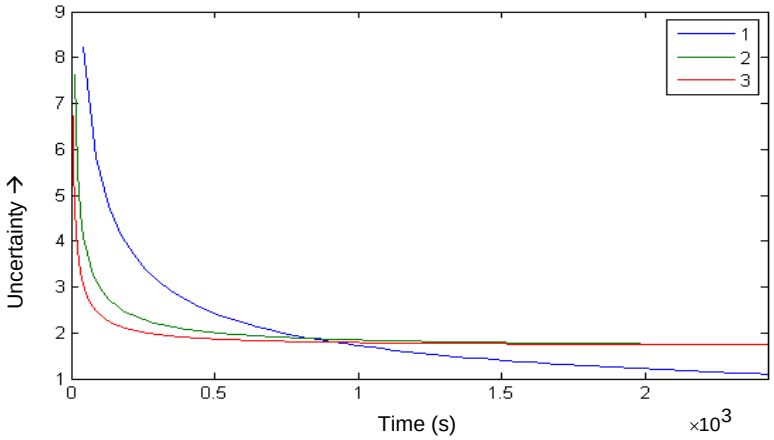

Figure 3: A decreasing uncertainty over time for three models of varying complexity. Here, the total uncertainty is shown for models 1-3 for any time budget between 0s and 2,500s.

Multiple Deterministic Instances

In the second set of experiments, we compare the reliability of multiple strategies for selecting appropriate instances and models. Recall that the objective here is to select those instances and models that minimize the total uncertainty on the whole set of instances given limited time. We use two models here, i.e. a reference model (Model 1) and one approximate model (Model 2). Models are generated by the procedure outlined at the beginning of this section. In order to determine reliability, we first calculate the real outcomes for all instances using the reference model.

In line with the approach outlined above, we compare the following eight instance selection strategies.

Strategy	Description
Model 1	Every evaluation is done with the reference model on a random untested instance.
Model 1 (int)	This strategy is the same as above, but in addition interpolation is used to predict the outcomes and uncertainties for those instances that have not been evaluated yet.
Model 2	Every evaluation is done with the approximate model on a random untested instance.
Model 2 (int)	This strategy is the same as above, but in addition interpolation is used to predict the outcomes and uncertainties for those instances that have not been evaluated yet.
Online	The online optimization strategy mentioned before, which selects the combination of instance and model that is expected to best reduce uncertainty over time.
Offline (800s)	The offline optimization strategy mentioned before, which calculates a schema of evaluations (models and instances) that is expected to reduce the uncertainty the most at the end of a given time budget, in this case 800 seconds.
Offline (1,800s)	The same strategy is also run with a time budget of 1,800 seconds.
Offline (3,800s)	Finally, the offline strategy is also run with a time budget of 3,800 seconds.

Figure 4 shows the reliability of each of the eight strategies over time. Reliability is measured here for each instance as the normalized difference between the real outcome for this instance, and the given strategy's outcome. Normalization is performed such that a reliability of 1 means that the strategy gives exactly the real outcome, whereas a reliability of 0 means that the strategy obtains no outcome or an outcome with an infinitely sized confidence interval. For some more intuition behind the reliability, we remark that in our Navy use case, a reliability of 0.85 indicates that an outcome has an uncertainty of approximately 5km, which is indeed a rather acceptable uncertainty according to our subject matter experts. We average the reliability over all instances and report results over time.

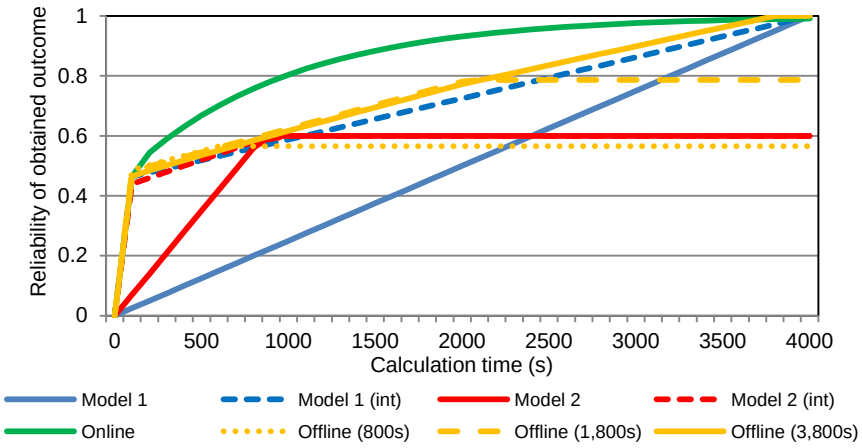

Figure 4: Reliability of all instance selection strategies over time.

The two solid red and blue lines represent the traditional approach of choosing a single model and evaluating the outcomes after all runs are performed. The two dotted red and blue lines still represent the reliability of an approach where a single model is chosen. However, for instances that have not yet been evaluated, interpolation is used to generate results. From the results in Figure 4, it is clear that by interpolation, we will gain good estimates of the outcome without evaluating all instances even if we decide to use only one model. The dotted red and blue lines clearly lie above the solid red and blue lines for every time budget, until there is enough time to evaluate all instances with the given model. Obviously, interpolation will not provide any benefit if all instances can be evaluated. Therefore, the dotted and solid red and blue lines end at the same level. We note that the approximate model (Model 2, red lines) clearly cannot end up at a reliability of 1 on the vertical axis, since it will yield a difference with reference model outcomes even if all instances can be evaluated.

The more effective strategies in Figure 4 are the online and offline ones discussed in the Approach section. The online strategy was tested once. The offline strategy was tested three times, i.e. with total time budgets of 800s, 1,800s and 3,800s. When we calculated the true outcomes, we found that every instance can be evaluated with the reference model in about 4,000s; therefore a choice for time budgets that are relatively small (800s), moderate (1,800s), and large (3,800s), respectively. More precisely, since running the approximate model on all instances takes 1,000s and running the reference model takes 4,000s, there is a need for selecting the appropriate instances and models for many of these time budgets.

From the previous section, the attentive reader may recall that we expected the offline strategy to outperform the online strategy, since the online strategy has no notion of the available time. However, the results show otherwise; the online strategy performs better throughout the available time budget, except for the last two seconds. The only possible reason for this is that the information (expected model outcomes and uncertainties) used in the offline process was imprecise and therefore a suboptimal evaluation schema was created. The online strategy on the other hand was able to adapt to the new information acquired during the process and therefore made much better progress. Only in the end it was outperformed by the offline method since it wasted some time by evaluating some instances with both models. Therefore, which of the two strategies is more effective depends on the available information in the initial example database. The better the information, the better the offline strategy will perform. Having said this, we note that both strategies significantly outperform a single-model-strategy on accuracy Furthermore, both have the added benefit of not requiring any difficult choices from the analyst.

Use Cases

The results presented above have been derived using synthetic data. However, the algorithms have also been tested on the use cases presented earlier. Given the sensitivity of the obtained results, we are not able to provide a detailed overview. Therefore only brief insights are given here.

For the first use case an example database was created containing results for various sonar models of different fidelity. We observed that the algorithm is capable of determining the best model to use given any available amount of time. It clearly helps analysts to not only see an approximate outcome, but also an indication of the reliability of this outcome given limited time. Furthermore, we identified a number of models in our study that had no added value because they did not provide the most reliable result for any time budget. Thus, the proposed algorithm helps in selecting those models that are able to contribute and discarding those that are not.

For the second use case, a different example database was created containing the features of interest for various seabed types to be encountered in the region of interest. Results for two different sonar models were also stored in this database. In order to find the sonar detection ranges within a grid of points and directions (these were the instances), the approach was employed using baseline as well as improved selection strategies. The results we obtained were similar to the synthetic results presented above. The overall conclusion here was that our improved selection strategies resulted in a significantly higher reliability of the outcome at any given time compared to the baseline selection strategies (running a single model). The offline strategy, as in the synthetic experiment, suffered from a relatively underspecified initial example database and therefore performed even less convincingly than in the synthetic experiments. On the other hand, interpolating outcomes and uncertainties for instances that had not yet been evaluated provided a benefit similar to that outlined for the synthetic data.

Thus, our approach can give a more reliable estimate of the true outcome for all instances in a shorter amount of time than approaches that are currently often employed by analysts. Our approach is therefore capable of increasing the safety of the people involved in actual missions.

CONCLUSIONS

In this paper we have shown that the combined selection of models and instances has a strong influence on the reliability of a solution. We identified two factors leading to the need for multiple evaluations, i.e. stochastic variables in the instances that need to be evaluated and the requirement to evaluate multiple instances (but not all of them due to time constraints). For two problem settings, namely a single instance with stochastic variables and multiple deterministic instances, algorithms were created to automatically perform both model and instance selection.

For problems containing a single instance with a stochastic variable, the devised algorithm clearly shows that the most appropriate model strongly depends on the runtime constraints. The larger the time budget, the more accurate (and slow) the selected model will be. Thus, an analyst can select the best model to use given any time budget and can even discover that certain models have no added value because they never will yield useful results.

When the problem consists of having to evaluate multiple deterministic instances under time constraints, the devised approach performs automatic selection of both the model as well as the instances to evaluate. We have introduced two different strategies that can be used within the algorithm (an online optimization strategy and an offline optimization strategy) and compared results on a synthetic test set and on a Naval use case. Offline optimization of model and instance selection can outperform an online optimization strategy when the available calculation time budget is known in advance and there is a sufficient amount of data on the models' relative performance. Otherwise, it is best to choose the online optimization strategy, as this can adapt on the fly to incoming data, at the expense of regularly evaluating the same instance multiple times with multiple models.

In general, we show that selecting the appropriate model and instances is a hard problem and argue that human intuition often yields suboptimal outcomes. The baseline strategies discussed in this paper are similar to those generally chosen by human analysts and are convincingly outperformed by our algorithms. We can conclude that both algorithms are very successful at addressing the problem at hand. Furthermore, since both algorithms are very modular, combining them is easy; therefore we can create an algorithm that can select which instances to test, with which model and with how many evaluations, in the most optimal manner.

Future work can follow a number of distinct directions. First, the approach presented here can be improved by integrating automatic feature detection (e.g. Jones, 1995) rather than manually defined features. Second, the two algorithms presented can be combined, in a rather straightforward manner, to cover the situation where multiple instances with stochastic variables need to be evaluated. Third, the approach can be applied in operational settings. At the time of writing, a collaboration with the Dutch Navy on this topic is in preparation.

REFERENCES

F. Hutter, H. Hoos, K. Leyton-Brown. Algorithm Runtime Prediction: The State of the Art. CoRR, abs/1211.0906 (2012)

J.R. Rice The algorithm selection problem Adv. Comput., 15 (1976), pp. 65–118

M. Roberts, A. Howe. Learned models of performance for many planners. ICAPS 2007 Workshop AI Planning and Learning (2007)

L. Kotthoff, I.P. Gent, I. Miguel. An evaluation of machine learning in algorithm selection for search problems AI Commun., 25 (3) (2012), pp. 257–270

E. Fink. How to solve it automatically: Selection among problem-solving methods. Proceedings of the Fourth International Conference on AI Planning Systems, AAAI Press (1998), pp. 128–136

M. Gagliolo, J. Schmidhuber. Dynamic algorithm portfolios. International Symposium on Artificial Intelligence and Mathematics (ISAIM'06) (2006)

K. Smith-Miles. Cross-disciplinary perspectives on meta-learning for algorithm selection. ACM Comput. Surv., 41 (1) (2009), pp. 6:1–6:25

Chiles, J.-P. and P. Delfiner (1999) Geostatistics, Modeling Spatial Uncertainty, Wiley Series in Probability and statistics

A.E. Howe, E. Dahlman, C. Hansen, M. Scheetz, A. Mayrhauser. Exploiting competitive planner performance, in: S. Biundo, M. Fox (Eds.), Recent Advances in AI Planning (ECP'99), Lecture Notes in Computer Science, vol. 1809, Springer, Berlin, Heidelberg (2000), pp. 62–72

T. Jones, S. Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. Proceedings of the 6th International Conference on Genetic Algorithms (ICGA'95) (1995), pp. 184–192

E. Weinberger. Correlated and uncorrelated fitness landscapes and how to tell the difference. Biol. Cybern., 63 (1990), pp. 325–336

Bartz-Beielstein, T. Sequential parameter optimization. Evolutionary Computation, 2005. The 2005 IEEE Congress on (Volume:1) 773 – 780

Matheron, G., "Principles of geostatistics", Economic Geology, 58, pp 1246–1266, 1963