Knowledge-guided Genetic Algorithm for input parameter optimisation in environmental modelling

Kerstin Wendt, Ana Cortés, and Tomàs Margalef

Departament d’Arquitectura de Computadors i Sistemes Operatius, Escola d’Enginyeria, Universitat Autònoma de Barcelona, 08193 Bellaterra (Barcelona), Spain

Abstract

The need for input parameter optimisation in environmental modelling is long known. Real-time constraints of disaster propagation predictions require fast and efficient calibration methods to deliver reliable predictions in time to avoid tragedy. Lately, evolutionary optimisation methods have become popular to solve the input parameter problem of environmental models. Applying a knowledge-guided Genetic Algorithm (GA) we demonstrate how to speed up parameter optimisation and consequently the propagation prediction of environmental disasters. Knowledge, obtained from historical and synthetical disasters, is stored in a knowledge base and provided to the GA in terms of a knowledge chromosome. Despite of increased loads of knowledge, its retrieval times can be kept near-constant. During GA mutation, ranges of selected parameters are limited forcing the GA to explore promising solution areas. Experiments in forest fire spread prediction show how time-consuming fitness evaluations of the GA could be lowered remarkably to cope with real-time capabilities maintaining the error magnitude.

Keywords: input parameter optimisation, environmental modelling, knowledge-guided Genetic Algorithm, forest fire spread prediction

1. Introduction

There exists a wide range of environmental models to simulate and predict complex natural phenomena. These comprise models for simulations in meteorology, oceanography, groundwater hydrology, and petroleum reservoirs as well as models predicting the propagation of disasters as floods, tsunamis, hurricanes, and forest fires. Common to all of them is that the precision of simulation output heavily depends on the quality of the provided input parameters. Especially applications implementing models for the prediction of natural catastrophes require most reliable simulation outcomes. Furthermore, these applications often have to fulfil stringent real-time constraints to be of use during an ongoing disaster. Because most simulation software works with well-founded and widely accepted models, the need for input parameter optimisation to improve model output is a long-known and often-tackled problem. Particularly

\*This work was supported by the MEC-Spain under contract TIN 2007-64974.

Email address: kerstin.wendt@caos.uab.es (Kerstin Wendt)

Corresponding author

1877-0509 © 2012 Published by Elsevier Ltd. Open access under CC BY-NC-ND license. doi:10.1016/j.procs.2010.04.152
in environments where correct and timely input parameters cannot be provided, efficient computational parameter estimation and optimisation strategies are required to minimise the deviation between the predicted scenario and the real phenomenon behaviour.

Many approaches for parameter calibration mainly use standard non-linear function optimisation techniques, which are not fully capable of handling high dimensionality and irregularities contained in environmental models [1]. With the continuously increasing availability of computing power evolutionary optimisation methods, especially Genetic Algorithms (GA), have become more popular and practicable to solve the parameter problem of environmental models [2, 3, 4, 5, 6].

In this paper, we formalise a technique to enhance the propagation prediction of environmental disasters. The case of forest fire spread prediction is chosen to demonstrate how the application of a knowledge-guided GA is suited to optimise high-dimensional and highly dynamic model input during real-time prediction using a knowledge base of considerable size.

The remainder of this work is organised as follows. The next section explains the sources for imprecision in input parameters and gives an overview of forest fire spread prediction. In section 3, we describe in detail how domain-specific knowledge can be used in a GA for parameter optimisation. Experimental results are shown in section 4 and section 5 outlines main conclusions and briefly discusses future work.

2. Real-time prediction of disaster propagation

2.1. The input parameter problem

Imprecision in the normally large number of input parameters is a known and serious problem in environmental modelling, because model output is directly affected delivering unreliable predictions. In disaster propagation models input parameters often are difficult or even impossible to measure. Although advances in power supply for sensors have been made recently [7], installation and maintenance of sufficient sensors remain a fundamental hindrance in areas difficult to access. Instead, data samples provided by maybe distant meteorological stations are used, but might offer measurements for uncommon parameters (e.g. fuel moisture) too seldom to be of use during an ongoing hazard prediction. Furthermore, many parameters are highly dynamic and tend to change rapidly in the microclimate generated by a disaster (e.g. strong wind gusts in forest fires). Additionally to imprecision in input parameters, grid-based propagation models have to deal with spatial uncertainty [8]. When real-time constraints have to be met, cell size for the region under consideration is often chosen rather big to reduce simulation runtime and consequently the contained heterogeneous environments (e.g. different vegetation models) might not be represented with the necessary degree of exactness.

Although correct parameter values became available as the prediction advances, often simulators work with a static set of input parameters and offer no ability for data assimilation during a running simulation. And because only the minority of models and simulators includes parameter calibration techniques by default, the work with outdated field measures, estimated, extrapolated and missing values remains generating unsatisfactory results. This is why we propose a general and simulator independent approach to enhance the prediction of disaster propagation by using a knowledge-guided GA for input parameter optimisation, always following requirements are met: (1) The range of each input parameter is known and defined, (2) The sensitivity index of each model parameter is known or at least the most sensitive input parameters are identified, and (3) Information about the true state of the disaster is available in a reasonable time interval.

2.2. Forest fire spread prediction

Where forest fires are not a natural part of the ecosystem and where they cannot be prevented, their prediction has become a crucial topic. Forest fires not only damage important ecological resources but also count for a global source of emissions to the atmosphere, provoke climate changes, air pollution, erosion, floods and water contamination, destroy buildings and infrastructure and, maybe most importantly, claim deaths. To avoid tragedy, effective tools are needed to quickly and reliably forecast the propagation of an ongoing fire. For that purpose mainly computer modelling and simulation applications are used to predict the fire front for a given time in the near future. There exists a large variety of wildfire behaviour and propagation simulators (e.g. FARSITE, BehavePlus, fireLib, NEXUS), many of them based on the Rothermel equation model [9].
These simulators traditionally work with a set of input parameters describing the environmental conditions of the region where the fire takes place including vegetational, climatological and topographical characteristics. The classic way of predicting forest fire behaviour takes the initial state of the fire front (RF = real fire) as input as well as the input parameters given for some time \( t_x \). The simulator then returns the prediction (SF = simulated fire) for the state of fire front at a later time \( t_{x+1} \). Figure 1 (a) summarises this process.

Comparing the simulation result SF from time \( t_{x+1} \) with the advanced real fire RF at the same instant, the forecasted fire front tends to differ to a greater or lesser extent from the real fire line because the calculation of the simulated fire is based upon a single set of input parameters afflicted with the before mentioned insufficiencies. Therefore, a data-driven prediction scheme was proposed to calibrate model input parameters applying a Genetic Algorithm [10, 11]. Introducing a previous calibration step as shown in figure 1 (b), the set of input parameters is refined before every prediction step. Different combinations of input parameter values are generated and evaluated. Comparing each simulated fire front at time \( t_{x+1} \) to the real fire front at the same time the quality of the corresponding parameter set can be obtained. Lastly, the best fitting scenario is selected to serve as input for the following prediction step. The objective is to find a simulator input that would describe best current environmental conditions. Having detected these values, it is argued that the same set of parameters could also be used to describe best the immediate future. Thus, the prediction now becomes the result of a series of automatically adjusted input configurations.

Although the data-driven prediction scheme significantly enhances the quality of input parameters [11], it obviously adds a non-negligible amount of computational effort, which should be reduced to a minimum in real-time disaster modelling applications. A strategy to combat this drawback using domain-specific data stored in and retrieved from a knowledge base to guide and speed up the GA is formalised in the next section.

### 3. Parameter optimisation with knowledge-guided Genetic Algorithm

Short simulator response time is a key characteristic to cope with real-time capabilities and that simulation outcomes can be of use during an ongoing disaster. Certainly, input parameter optimisation based on GA improves prediction result quality but introduces an additional computational burden at the expense of prediction runtime. To deliver satisfactory results the GA has to run for a high number of generations or evaluate a huge population to find an input scenario which generates predictions with sufficient precision. Furthermore, environmental modelling problems tend to possess a large quantity of input parameters creating a vast search space to be explored by the GA. Although applying parallel computation in order to reduce global computing time, parameter calibration is not yet feasible for large real cases.

One proposal to overcome these difficulties is to guide the optimisation process towards promising solution areas using domain-specific knowledge [10, 12]. In doing so, a faster convergence of the GA towards fitter solutions can be reached. In recent years, it proved of value for certain problems in different areas to guide GA introducing problem-specific knowledge [13, 14, 15, 16, 17], where a special interest in the use of non-random mutation operators could be observed. In general, the quality of the optimisation is remarkably better when domain knowledge is incorporated in the problem solving process [18].

Up-to-date, few general principles, guidelines and best practices on how to incorporate which type of knowledge into GA exist and the efficiency of this method is mainly proven by experiments. This is why current investigation tries...
to generalise the introduction of domain knowledge in evolutionary algorithms [18, 19, 20]. However, the approach of using additional information results fully legitimate and reasonable as it tries to imitate behaviour and experience of human system experts. Like domain experts applying their knowledge of observed phenomenon behaviour to rectify decision support from automated prediction systems when forecasting a new disaster, we investigated the insertion of domain-specific knowledge into the GA. In doing so, we will be able to ensure that the input parameter optimisation process does not become a bottleneck of the overall prediction. Figure 2 shows an overview of the enhanced data-driven prediction method inserting domain data during the GA.

Because reasonable growth of knowledge is expected in the near future due to improved observation technologies and documentation possibilities, it is necessary to find the corresponding knowledge in a very efficient manner to avoid a new increase in prediction runtime. In the following, our approach is explained in detail.

3.1. Knowledge-guided Genetic Algorithm

Genetic Algorithms are a heuristic optimisation method and can be used to find exact or near-optimal solutions to search problems. Their functionality is inspired by evolutionary biology and they repetitively apply the methods elitism, selection, crossover, mutation, fitness calculation, and reinsertion. GA can be an effective tool for parameter optimisation in environmental modelling [1], but their elevated execution time and complex parameter tuning have limited their deployment in real-time disaster prediction.

The GA used for parameter calibration in the data-driven forest fire prediction works on a population $x = [x_1, x_2, ..., x_p]$ consisting of $p$ individuals (chromosomes) $x_k = [x_{k1}, x_{k2}, ..., x_{kd}]$, $\forall k = 1 \leq k \leq p$. Each individual of the population represents a fire simulator input parameter set (scenario) made up of $d$ parameter values for e.g. fuel model, slope, fuel moistures and wind characteristics. The genes $x_{kj} \in [lb_j, ub_j]$ of each individual are encoded as real values within a previously defined range, where $lb_j, ub_j \in \mathbb{R}, \forall k = 1 \leq k \leq p, j = 1 \leq j \leq d$ and $lb_j$ being the lower bound for parameter values for gene $j$ and $ub_j$ the upper bound for gene $j$.

The goodness of the generated scenarios is evaluated by a problem dependent fitness function. In the fire prediction context, the fittest scenario is the one that generates a simulated fire map the most similar to the real map of fire propagation. To determine the fitness of each scenario, an error function (1) based on a cell-by-cell comparison of the affected terrain is applied putting into relation the erroneous burnt cells of the simulated fire with all really burnt cells.

$$\text{error} = \frac{(\bigcup -\text{initial\_fire}) - (\bigcap -\text{initial\_fire})}{\text{real\_fire} - \text{initial\_fire}}$$

with $\bigcup$ resulting in the number of cells burnt in one or both real and simulated fire map and $\bigcap$ denominating the number of cells burnt in both real and simulated fire map is to be minimised during the optimisation process.

Particularly, the fitness evaluation for each scenario is a very time consuming procedure and adversely affects speed and efficiency of the overall calibration, a serious issue in real-time disaster modelling. For every individual $x_k$ in the population of the GA a simulation has to be run to be able to compute the goodness of this individual.
The objective thus is to reduce the number of fitness evaluations to a practicable minimum during GA operation. Decreasing the number of individuals in the population will not be considered as this approach normally leads to an unwanted loss of population diversity and hence to an more unfit final result. To provide near-optimal solutions with sufficient precision in early generations, i.e. yield short convergence times of the GA, we consider embedding domain knowledge into the GA to quickly guide it towards promising solution areas.

3.2. Knowledge retrieval

To make proper use of expert knowledge during GA execution we built a knowledge base (KB) containing information about historical real fires and synthetical simulated fires. In a Relational Database Management System (RDBMS) we store model input parameter values, e.g. details on fuel model, slope, wind properties, together with their corresponding model output parameters, e.g. speed and direction of fire spread. This approach does not require the complete set of input parameters to be stored, but best results are obtained if the most sensitive and dynamic parameters are available.

Knowledge is then retrieved from the KB once in every calibration step before starting the GA: Model output parameters (maximum fire spread direction and speed) are obtained from the real fire propagation at time $t_{i+1}$ and taken as arguments to query the KB. To speed up the knowledge retrieval process available static input parameters (e.g. slope, fuel model) have been added as query arguments. Given a specific phenomenon behaviour, we can then find its causing dynamic conditions. The query result delivers the input parameter values from forest fires that performed the most similar compared to the ongoing event. The encountered values are then gathered in a knowledge chromosome $k_c = [k_{c_1}, k_{c_2}, ..., k_{c_d}]$ which is stored temporarily and thus available in every phase during each iteration of the GA.

In order to find the $k$ most similar event parameterisations we carry out a $k$-Nearest Neighbour search. We apply a distance function based on the Heterogeneous Euclidean-Overlap Metric (HEOM) [21] to measure similarity between individual events. This distance measure takes into account differences between nominal and linear parameters: For nominal data a value-matching-based metric is defined and linear parameters are compared with the normal Euclidean metric including range normalisation to avoid that parameters with large ranges overpower those with smaller ranges.

The sensitivity analysis conducted by [10] shows that fire spread is influenced to varying degrees by the existing input parameters. In order to correctly reflect this sensitivity of inputs and to obtain meaningful results we added HEOM a parameter importance ranking by means of a weight factor

$$w_k = \frac{d - (i(k) - 1)}{\sum_{i=1}^{d} s}$$

(2)

with $w_k$ being the weight of the $k^{th}$ query argument, $d$ the number of dimensions in the $k$-NN search space and $i(k)$ the importance number of the $k^{th}$ query argument (e.g. $i(fuelmodel) = 1$, $i(slope) = 2$) with $1 \leq i \leq d$ and the smallest importance number representing the highest impact on model output. Our distance measure finally results in

$$WHEOM(\vec{x}, \vec{y}) = \sqrt{\sum_{k=1}^{d} w_k (x_k - y_k)^2}$$

(3)

where $h_k$ stands for the two mentioned metrics.

3.3. Knowledge insertion

The retrieved knowledge is then used during mutation to guide the GA towards promising solution areas. In nature, mutation occurs very infrequently and can often result in a weaker individual. Occasionally, the result might be to produce a stronger one. In GA, mutation is an operator that changes the information contained in one or more gene values in an individual according to the defined mutation probability. This probability usually is set fairly low. Directing the mutation process and transforming randomness into controlled variation, a significantly increased mutation probability up to 0.4 should be considered. Even more, as stronger individuals are ensured and we require a fast convergence.

The most common mutation operator for real-valued parameters is the uniform mutation that replaces the value of a chosen gene $x_{kj}$ with a uniform random value selected from the problem-specific parameter range between lower
and upper bound \([lb_j, ub_j]\). Our guided mutation approach uses the domain knowledge contained in the knowledge chromosome \(kc\) to narrow parameter ranges. Parameter values then oscillate in their smaller ranges, finally, forcing the GA to adopt specific values in certain dimensions from which we know that they will increase an individual’s fitness. The three steps to follow for each gene during mutation are:

**Preparation.** Compute the mutation probability for the gene by associating a random number from the interval \([0, 1]\) with the gene. The gene is mutated if the associated number is less than the specified mutation rate.

**Knowledge insertion.** If knowledge is available for this gene in the knowledge chromosome \(kc\), then re-define the gene’s range of valid parameter values. Set the new lower bound \(lb_{r_j}\) to

\[
lb_{r_j} = kc_j - t_j
\]

and the re-defined upper bound \(ub_{r_j}\) to

\[
ub_{r_j} = kc_j + t_j.
\]

If necessary, the bounds have to be repaired after knowledge insertion in order to obtain a consistent program: If \(lb_{r_j} < lb_j\) then reset \(lb_{r_j} = lb_j\), and if \(ub_{r_j} > ub_j\) then reset \(ub_{r_j} = ub_j\).

**Mutation.** If the gene needs to be mutated the gene is modified choosing a random value from the original range, or, if applicable, from the narrowed range of valid values.

The configurable threshold \(t_j\) can be chosen independently for every gene. A valid subset of the original range is chosen by adding margins to the retrieved knowledge, as the knowledge retrieval process only returns parameterisations most similar to the real event. Depending on the degree of detailedness of the information in the KB, the configuration of the real fire might be found or not.

At present, we employ knowledge for wind speed and wind direction and cut the ranges for these two parameters because sensitivity analysis has proven that the input parameters which mainly affect fire propagation are wind and slope characteristics [10]. The present thresholds used as margin are 5 deg for wind direction and 2 mph for wind speed. There exist other factors influencing fire spread (e.g. fuel moisture, fuel type) but the rate of spread indicates far less sensitivity to these parameters.

### 4. Experimental evaluation

The experimentation’s objective is to prove the benefit of domain-specific knowledge. If inserted during mutation to guide the GA, we can reduce the number of generations performed by the GA and thus are able to decrease time-consuming fitness evaluations and cut in half parameter optimisation runtime while maintaining the prediction error magnitude. We compare the guided optimisation methodology with the data-driven prediction scheme not including any guidance during calibration for two different real fire plots. Further on, we demonstrate the effectiveness of our proposed knowledge retrieval approach presenting retrieval times for different implementational approaches of \(k\)-NN using WHEOM.

For the following test scenario we worked with the forest fire spread simulator fireLib [22] and two real fire plots which can be seen in figures 3 (a) and (b) showing the fire front evolution through time. Plots 751 (line ignition) and 533 (point ignition) are part of a set of prescribed burns performed in the frame of the European SPREAD project [23] in Gestosa (Portugal) in 2004 and measure 20 x 30 m and 95 x 123 m, respectively.

We used randomly initialised populations of 50 individuals for the GA. In order to get around arbitrary evolution effects and to get more descriptive results, the presented experiments are the averaged outcome of 10 different initial populations. During GA, 2 or 5 iterations were performed and we applied a mutation rate of 0.2 when guiding the mutation process, otherwise the mutation probability was 0.01.

Figure 4 (a) shows the input parameter refinement during calibration stage for plot 751 applying guided and non-guided mutation. It can be observed that, incorporating domain-specific knowledge results in an equal or sometimes even smaller calibration error. Using the optimised input parameter (the best individual after 2 or 5 generations) for the subsequent prediction step, we can note in figure 4 (b) that the parameter obtained after 2 generations having
inserted knowledge generates predictions with the same error magnitude as the parameter obtained after 5 generations without inserting domain knowledge. Figures 5 (a) and (b) show a similar behaviour for plot 533.

The knowledge retrieval and insertion process only adds negligible overhead to the total optimisation runtime as it is performed once before starting the GA. The formerly introduced distance measure WHEOM used during $k$-NN to retrieve the most similar fire configurations from KB was implemented in different ways in plain SQL. The most efficient implementation uses temporal variables to avoid repeating range calculations for linear parameters and the value-matching-based metric for nominal parameters was realised by means of WHERE-conditions. This prevents $k$-NN from calculating the distance from the query point to every other fire parameterisation in the KB and thus significantly reduces the number of distance evaluations when additionally using a common index structure on nominal parameter columns. Requiring an exact match of the query argument value with the corresponding nominal parameter value also prevents $k$-NN from returning erroneous results and eventually misguiding the GA. Thus, to some extent, we are able to control the influence of knowledge quality on the performance of the GA.

Figure 6 shows the evolution of retrieval times applying different retrieval implementations and steadily increasing the amount of knowledge contained in the KB (starting with approx. 12,500 records). For all test cases only retrieval times were measured disregarding the time needed for connection or disconnection in case of the database or the time required to open and close the text file in case of the text-based approach. It is clearly visible that the performance of knowledge retrieval highly depends on its specific implementation. Simply storing the data in a relational knowledge base (KB) without making any use of database specific enhancements hardly yields faster retrieval times compared to employing a formerly used text file-based approach (text-based) [12]. Enhanced retrieval times can be obtained when value matching of nominal parameters is implemented during distance evaluation using WHERE-conditions (KB + WHERE-cond.). Applying WHERE-conditions combined with indices (KB + WHERE-cond. + indices) yielded the
Figure 5: (a) Calibration step and (b) prediction step results for plot 533.

Figure 6: Comparison of knowledge retrieval times applying different storage and retrieval methods.

most important time savings with a nearly constant retrieval time though increasing the amount of knowledge. These experiments show that the applied knowledge guidance method does not have any harmful effects on the overall optimisation process.

5. Conclusions and future work

In this paper, we have formalised an approach for input parameter optimisation for real-time disaster modelling using the example of forest fire spread prediction. The optimisation process takes advantage of domain-specific knowledge guiding a GA during mutation towards promising solution areas notably speeding up the calibration of highly dynamic or unavailable model input. This gain in optimisation speed helps the simulation application to cope with real-time requirements and to deliver predictions closer to reality for online decision support.

Future work includes determining the benefit of including additional domain-dependent information (e.g. fire shape) into the KB and to prove the potential of the approach with a different fire simulator or another disaster application. Possible GA enhancements comprise investigating on knowledge incorporation in further GA phases and the use of decreasing parameter ranges during knowledge insertion. The application of intelligent paradigms (e.g. clustering algorithms, fuzzy inference systems) together with Evolutionary Algorithms is also a promising research field which we will continue studying.

References


