18th World IMACS / MODSIM Congress, Cairns, Australia 13-17 July 2009
http://mssanz.org.au/modsim09

Calibration of hydrological model parameters with the JAMS framework

C. Fischer[1], S. Kralisch[1], P. Krause[1], M. Fink[1] and W.-A. Flügel[1]

[1] Department of Geoinformatics, Hydrology and Modelling,
School of Chemical and Earth Sciences
Friedrich-Schiller-University, Jena, Germany
E-Mail:christian.fischer.2@uni-jena.de

Abstract:

In order to apply environmental models successfully, it is essential to determine model parameters very carefully. A direct determination of parameter values is often not feasible since they lack a clear physical meaning or field measurements are too expensive. Therefore, estimation of parameters is usually done by fitting model response and observation in a trial and error process. Depending on the model's complexity and the number of calibration parameters, this process is often too time-consuming.

The Jena Adaptable Modelling System (JAMS) is a framework for component based model development and application, which was designed to meet current challenges in sustainable management of water resources. This task demands for integrated, flexible and easy to use environmental simulation models, which are able to simulate the quantitative and qualitative aspects of the hydrological cycle with a sufficient degree of certainty.

Model calibration in JAMS is supported by a semi-automated assistant who guides the user step by step through the process of setting up the calibration procedure. The uniform structure of JAMS models allows for easy analyses and modifications of the model. This includes (i) automatic detection and removal of model components, which are irrelevant for model calibration, (ii) rearranging model components to speed up optimization and (iii) assistance in the parameterization of the optimization method.

In this paper, three optimization methods are presented and compared to each other in order to depict our procedure. The first one is the Shuffled Complex Evolution algorithm (SCE). This evolutionary method has been used for hydrologic model calibration for some years. Branch \& Bound, a classical method based on properties of the objective function and with some interesting theoretical characteristics, is presented as second procedure. Finally, a radial basis function method is presented as a representative of the class of response surface based methods. This method is especially interesting for optimization tasks that involve high runtimes for single model evaluations and have shown very good results in optimization benchmarks.

Keywords: *Parameter Calibration, Optimization, Modelling Frameworks, Environmental Modelling*

1. INTRODUCTION

The Jena Adaptable Modelling System (JAMS) is a modular structured environmental modelling framework which has been developed to meet current challenges in sustainable management of water resources. For this purpose JAMS provides special data types that can be used to describe the spatial and temporal domains. For flexibility and extensibility the interior control flow is not fixed but can be configured from outside, e.g. with special software components. It features functions for management and manipulation of environmental data, e.g. for reading and writing time series data or for unit conversion (Kralisch and Krause, 2006)

During the last years a number of JAMS simulation components have been developed that cover various aspects of integrated water resources management, particularly hydrological modelling, nutrient modelling, parameter optimization and visualization of model results. An overview can be found in Kralisch et al. (2007).

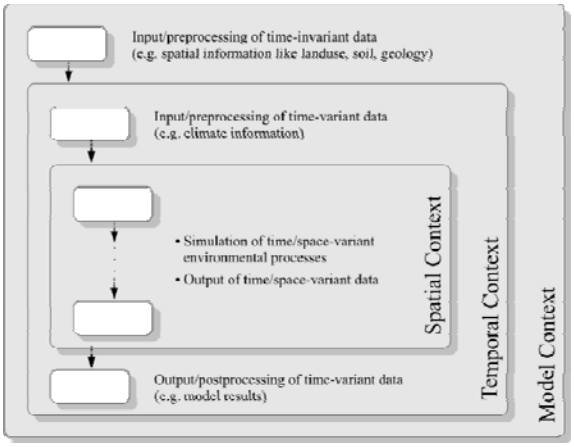


Figure 1. Common structure of spatial distributed environmental simulation models in JAMS

In order to apply these models successfully, it is essential to determine model parameters very carefully. A direct determination of parameter values is often not feasible since they lack a clear physical meaning or field measurements are too expensive. Therefore, estimation of parameters is usually done by fitting model response and observation in a trial and error process. Quality of parameter sets is usually quantified by error metrics like the Nash-Sutcliffe efficiency. This leads to a nonlinear single objective optimization problem. In practice, it is hard to find a best parameter set, because often the objective function is not smooth or even discontinuous, the feasible parameter space can consist of multiple large attraction areas, each containing numerous local optima. Local search methods can get stuck in them far from a global solution. Usually calibration runtime is scaling exponentially with parameter quantity. This is known as *Curse of dimensionality* (Bellmann, 1966). Modern environmental models can have many parameters, making optimization very time consuming. Altogether, it is neither advisable nor feasible to manually calibrate a complex model. However, to overcome this problem a lot of research work was spent in the development of efficient automatic calibration methods (Yapo et. al., 1998). The following sections will provide an overview about optimization techniques and their applications in JAMS.

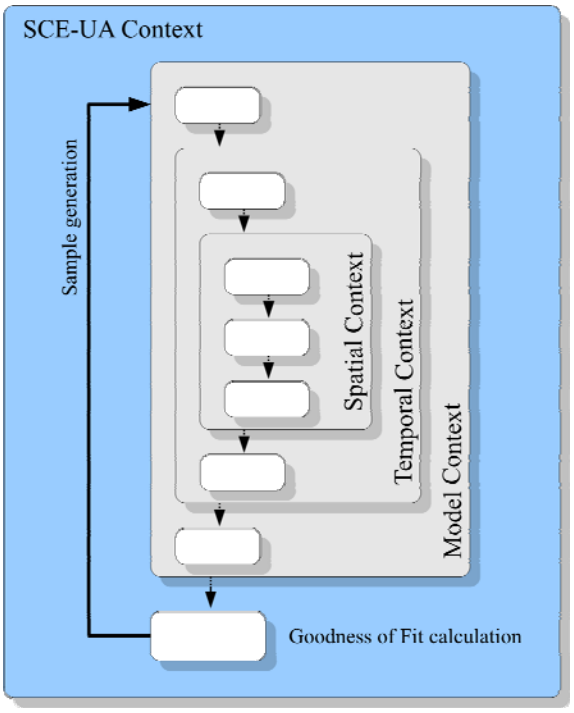


Figure 2. Model structure for search based parameter optimization

2. JENA ADAPTIVE MODELLING SYSTEM

JAMS have been developed with the main objective to create models that can simulate environmental processes at discrete points in time and/or space. This approach is widely-used by many distributed hydrological models applied in current practice. JAMS provides two specific types of building blocks, named *components* and *context components*. Components are used to implement specific knowledge as process algorithms whereas the model structure and the component's execution control is defined by the context components. As a benefit from this approach, an environmental process (e.g. potential evapotranspiration) can be implemented as a JAMS component without any knowledge about it's later execution context, e.g. the temporal resolution or the type of spatial discretisation of the modelled area. The only precondition for the later application

of a component in conjunction with others is the proper declaration of it's desired input data and provided output data by means of metadata in the component's source code. Context components in JAMS define the model structure that is needed to represent the necessary temporal and spatial iteration loops. This can be accomplished by managing the repeated execution of other components while varying their sets of input data. Each set of input data may then represent one point in time (*temporal context*) or space (*spatial context*). A context component in JAMS can thus be seen as a scope that defines an environment for the execution of other software components. Together with another context (*model context*) that manages the execution of the real simulation, the above-mentioned environmental models can easily be produced.

Figure 1 shows the different contexts and the workflow usually executed within them. In addition to the layout shown in this illustration, a JAMS model can include more than only one temporal or spatial context. As an example, a simulation model considering hydrological processes in the river channel could additionally include a second spatial context managing the iteration over river reaches and the execution of appropriate process components at a time.

JAMS calibration methods were implemented as context components. Due to the flexibility of the JAMS framework and the generality of the applied methods, only a minimum of knowledge about the model to be optimized is needed. Only the relevant model parameters, the objective function and method dependent parameters must be made available to the calibration component.

The resulting JAMS model structure is shown in figure 2. The calibration context component encloses the model to optimize, can access its parameter values and controls its execution. Since every search based optimization procedure comprises the two tasks search and evaluation, this setup depicts a general structure for any search based parameter calibration component in JAMS.

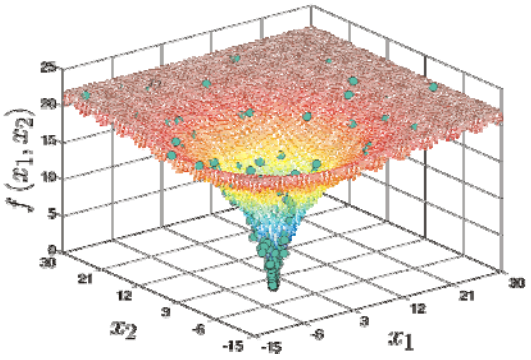


Figure 3. SCE-UA: Surface of the ackley function and the first 200 sampled points are marked in green

3. OPTIMIZATION METHODS

In the last two years some calibration methods were implemented for the JAMS framework. An overview about three selected optimizers is given here.

3.1. Shuffle Complex Evolution

The SCE-UA (Shuffle Complex Evolution - University of Arizona) algorithm was introduced by Duan et al. (1992). This evolutionary optimization method was developed especially for the application of parameter optimization in hydrological models.

The core idea of evolutionary optimization methods is to treat the optimization problem as a natural evolutionary process. Main subject of SCE-UA is therefore a population of samples - each of them representing one solution candidate. This population is divided into complexes that evolve independently from each other. In order to create new samples, sub complexes are formed which act as parents. While generally every sub complex is able to produce a new sample, promising sub complexes are preferred in this reproduction step. The produced children must fulfil some minimum requirements before they are added to the population and supersede the currently worst sample. After some iterations of reproduction, the complexes are joined. The process of complex segmentation and reproduction is repeated until no further improvement of the samples fitness can be accomplished.

In figure 3 the ackley function (Ackley, 1987) is shown. It is characterized by a very bumpy surface leading to many small local optima. The global optimum is located at zero. Points sampled by SCE-UA are marked green. Some of them are scattered all around the parameter space, these were mostly generated in the beginning when SCE-UA searches more globally. But the mass of dots is located near zero, because SCE-UA concentrates its activities near the optimum later on.

The SCE-UA algorithm shows good convergence for a vast variety of problems, meaning that with a sufficiently high (fixed) number of model evaluations the method has a fairly high probability to find the global optimum.

3.2. Branch & Bound

The class of Branch & Bound algorithms has a long history in mathematical optimization. The common strategy of these methods is to partition the problem into smaller sub problems (*Branch*), choose one of the sub problems and recursively apply this strategy. As soon as all sub problems have been solved, the solution of the original problem is also known.

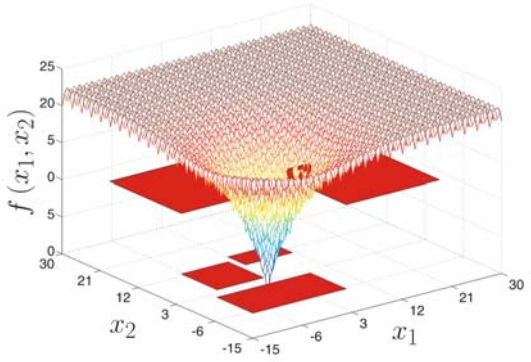


Figure 4. Branch and Bound algorithm: The red rectangles show the lower bounds of the sub problems after 5 iterations

Our implementation expects the feasible parameter space to form a d-dimensional rectangle. Subdivision is performed by bisecting the rectangle at its longest edge obtaining two new rectangles. If iterated over all sub problems, very poor runtime performance and huge memory usage will be the result. To avoid this, a strategy of rejection is needed. For each sub problem a lower bound is obtained by solving a relaxed problem. Based on the calculated lower bounds a promising sub problem is chosen in the next step. Sub problems with bounds worse than the best parameter set so far can be discarded. To obtain these bounds, *Lipschitz-continuity* of the objective function is assumed. For the approach, the *Lipschitz-constant* ***L*** must be known or at least approximated.

This process of branch and bound is recursively iterated. Under some weak assumptions, the discovery of the global optimum can be guaranteed. It can be shown that the value of the lowest lower bounds will converge towards the optimum. Hence, if the distance between the best sampled point and its lower bound is smaller than a prespecifed value, optimization will be stopped.

A detailed explanation of this algorithm is given by Horst et al. (2000). Figure 4 shows the behaviour of the Branch & Bound algorithm in respect to the ackley function. After five iterations the parameter space is partitioned into six rectangles. For every sub problem a fairly good lower bound was calculated, so that the rough position of the global optimum is already identified.

3.3. A radial basis function (RBF) method

This method is especially suited for optimizations with expensive function evaluations. Due to the strategy of surrogate based optimization, it is capable of finding a rough estimate of the solution within a few iterations in most cases. By interpolating already sampled points with radial basis functions, a surrogate of the objective function´s response surface is generated.

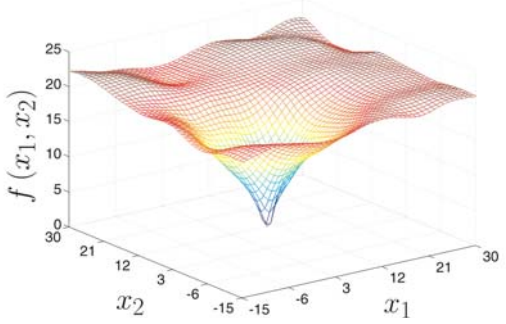


Figure 5. RBF procedure: Interpolated ackley function after 100 function evaluations

This helps to select parameter sets for evaluation very carefully. Before a new sample *x* is drawn, it is tested whether it is reasonable to assume an optimum there. Therefore, a hypothetic sample *x* is interpolated, which is slightly better than the best parameter set found so far. If this leads to an odd and bumpy interpolation, the assumption was not justified. On the other hand, a smooth interpolation gives reason for searching an optimum there. Based on that objective the algorithm searches the next candidates for evaluation by using a standard optimization routine. Thus the optimization algorithm itself needs a large amount of processing power. However, if function evaluations are very expensive this is negligible. A detailed description of the RBF method is given in Gutmann (2001).

It was also observed that the method finds the approximate optimum fast, but converges rather slowly. To avoid dispensable computing time the optimization is stopped after a fixed number of iterations and the nelder mead downhill simplex method is taking over to determine the exact location of the optimum.

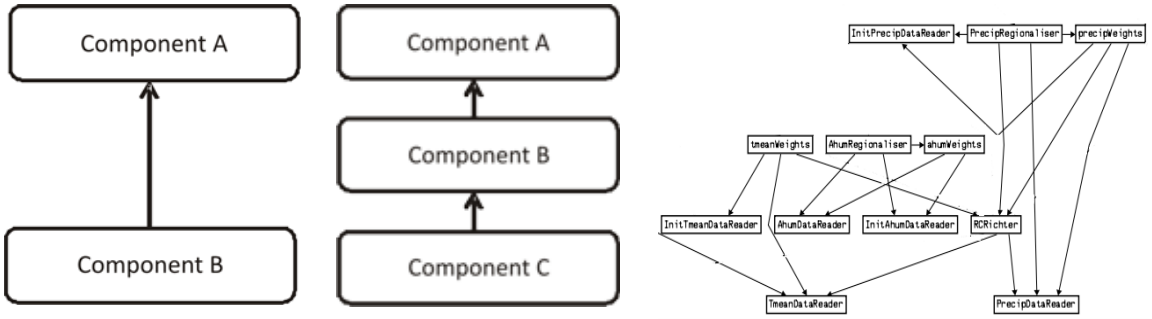
Figure 5 shows the interpolated surface of the ackley function after 100 samples drawn. The interpolation is much smoother than the original and the minimum about zero has not reproduced very well yet, but the general shape of the ackley function is clearly recognizable.

4. CALIBRATION ASSISTANT

For the calibration of JAMS-models, various adjustments of the model structure are desirable, e.g. selection, insertion and parameterization of the optimization component, removal of visual output components and selection of in- and output data. A software wizard provides support in this task and guides the user step by step through the process of calibration and gives assistance whenever possible.

4.1. Optimization procedure setup

As a first step of the calibration setup procedure, the model is loaded and the list of available parameters is presented to the user. Following, feasible value ranges of the selected parameters and one or more objective functions (e.g. Nash-Sutcliffe efficiency) need to be selected from a list. Based on this information, suitable optimization methods are proposed and parameterized with default values. For fine tuning, these parameterizations can be adapted. A comprehensive description of all values helps to choose well-performing parameters. Finally the software wizard automatically generates the calibration environment.



(a) Component B directly depending on A

(b) Component C indirectly depending on A

(c) Partial dependency graph of J2000

Figure 6. Different dependencies between components

4.2. Model adaption

In order to maximize runtime performance of the model and accordingly of the calibration procedure, all components that are not needed to ultimately compute the objective function can be removed. In order to identify those components, a dependency analysis of the model is being applied based on the components' input and output data. Figure 6 shows some examples. Assume that component A provides data, which need to be accessed by component B. Thus B is directly dependent of A. Assume another component C reading an attribute, which is written by B. Now C is indirectly dependent of A. Such a graph can easily be created for complex environmental models, e.g. for the model J2000, whose dependency graph is sketched in figure 6c.

Based on these graphs the model structure can be further analyzed. A component which is not connected with the objective function component does neither directly nor indirectly use any data generated by that component. That means such a component has absolute no effect on the objective and is therefore eliminated. Because every component provides information about their attributes this analysis is done fully automatically. Usually, these are components for visualization and data-output. In addition, components can be identified that have influence on the objective function, but are independent of the parameters in calibration, e.g. data-input processes. Under specific circumstances, these processes can be executed only once before calibration starts, thereby saving execution time. In order to make sure that each model starts with identical initial conditions, a snapshot of the complete model is being taken.. At each model execution in the course of the optimization procedure, this snapshot is being restored in advance.

5. PARALLELIZATION

The calibration task can be accelerated enormously by taking advantage of parallel computing systems. Therefore, a version of SCE-UA was developed, which is using GridGain (http://www.gridgain.com). GridGain is a Java based framework to support programming in distributed computing systems. It is suited for most architectures of parallel computing systems, but was especially designed for computing in grids. A grid is a collection of loosely connected computers, forming a powerful high performance supercomputer (Foster and Kesselmann, 2007).

Modellers, who want to calibrate a model, only have to choose the parallel SCE-UA optimization context. Everything else seems like it is running in a single process at the local computer. The framework searches independently for available nodes in the network and distributes incoming task as well as required resources among them. If a node fails to finish a task, it is detected automatically and the affected task is moved to another node. After execution output data is transmitted to a central node, where it is prepared for further processing.

SCE-UA manages one or more complexes during execution. The evolution of each complex is mostly independent of other complexes. This property allows simple parallelization of SCE-UA by allocating a computing-node for each complex. Communication between computing-nodes is reduced to synchronization and redistribution of complexes at the beginning and the end of each iteration.

The implementation was tested in a cluster environment consisting of 32 AMD Opteron Dual Core CPUs, where always four CPUs share their memory. The nodes are connected by a gigabit Ethernet connection. Sixteen nodes were assigned to GridGain. Accordingly, SCE-UA was started with sixteen complexes. In comparison to the serial execution, a very good speedup of 14 was reached, i.e. compared with the single machine optimization the GridGain procedure took only a 14th of its runtime. However this simple strategy is limited by the number of complexes. Further parallelization could be achieved with a modified nelder mead algorithm presented by Kolda (2006).

6. APPLICATION

The hydrological model J2000 allows the physically based simulation of the water balance in meso and macro scale catchments. It follows a distributed spatial approach and simulates the hydrological cycle on a daily basis. A detailed description is given by Krause and Flügel (2001).

To test the calibration methods presented in section 3, some modules of a J2000 model were selected for calibration. For this purpose the meso scale catchment of the Wilde Gera, located in middle Germany, was chosen. Table 1 shows the selected parameters under calibration and their feasible ranges. The Nash-Sutcliffe efficiency was selected as objective function. Parameterization of the calibration methods was easy, because the Branch & Bound and RBF methods do not possess any parameters and SCE was used with two complexes.

name	range	description
snow module		
t_{fac}	[0;5]	controls the linear effect of temperature on the potential snow melt rate
r_{fac}	[0;1]	determines increase of potential snow melt rate by rainfall
T_{Base}	[-5;5]	threshold for deciding whether precipitation is snow or rain
soil module parameters		
concRD1	[0;10]	retention coefficient for direct runoff
concRD2	[0;10]	retention coefficient for interflow runoff
VertLPS	[0;10]	calibration coefficient distributing the outflow from large pore storage to lateral interflow and vertical percolation.

Table 1. Selected calibration parameters of the J2000 model

Subsequent analysis of the model structure showed that the original model contained 61 components, 23 of which were identified as redundant for optimization. 8 of these 23 were GUI components, while the others were performing calculations which did not affect the runoff, e.g. temporal aggregation of model results. Furthermore, 14 components were identified to be completely independent of the parameters in calibration. They were executed only once before optimization begins. With these modifications, only 24 components stayed inside the calibration context. As a benefit from these modifications, model evaluation is done 8.8%

faster, but produces exactly the same results. These modifications could have been made by hand, but that would be fault-prone and would require much effort during setup.

The results of the application of the presented methods are shown in fig. 7. As can be clearly seen, all methods are able to calibrate the model and result in identical values of the objective function. The Branch & Bound method proved to be the fastest and found an optimum after only 100 iterations. The RBF method is also very fast in finding a good parameter set, but shows slow convergence and finally needed about 1000 iterations. Also SCE performs well. It finishes optimization after 500 iterations.

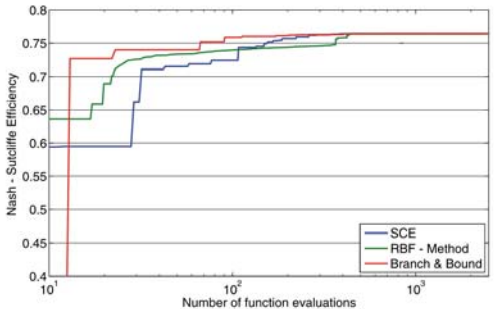


Figure 7. Progress of optimization in respect to function evaluations and method

7. CONCLUSIONS

This article gives an overview about calibration of environmental models with the JAMS - framework. The framework is described briefly and it is justified why calibration of environmental models is necessary and why automatic calibration methods are unavoidable nowadays.

Subsequently three optimization methods are presented, which have been implemented as JAMS contexts. These are the Shuffled Complex Evolution method, a Branch & Bound method and a radial basis function method. In case of the SCE method a parallel implementation was presented. It was shown how a calibration assistant can guide an inexperienced user through the setup of the calibration process. Additionally, this assistant is able to analyze the model structure and perform modifications for faster model execution. An application of the presented optimization methods showed that every method perform well. Modifications of the model structure lead to an enormous improvement; model execution time was reduced by 8.8%.

ACKNOWLEDGMENTS

We would like to thank the German Federal Ministry of Education and Research for funding the JAMS development in the course of the program "Entrepreneurial Regions - the BMBF Innovation Initiative for the New German Länder".

REFERENCES

Ackley, D. (1987), A connectionist machine for genetic hillclimbing. *Kluwer Academic Pub*.

Bellman, R. (1966), Dynamic programming. *Science*, 153 (3731):34–37.

Duan, Q., S. Sorooshian, and V. Gupta (1992), Effective and efficient global optimization for conceptual rainfallrunoff models. *Water Resources Research*, 28(4):1015–1031.

Foster, I. and C. Kesselman (2004), The grid: blueprint for a new computing infrastructure. *Morgan Kaufmann.*

Gutmann, H. (2001), A Radial Basis Function Method for Global Optimization. *Journal of Global Optimization*, 19(3):201–227.

Horst, R., P. Pardalos, and N. Thoai (2000), Introduction to Global Optimization. *Kluwer Academic Pub*

Kolda, T. (2006), Revisiting asynchronous parallel pattern search for nonlinear optimization. *SIAM Journal on Optimization*, 16(2):563–586.

Kralisch, S. and P. Krause (2006), JAMS – A Framework for Natural Resource Model Development and Application. In *Proceedings of the iEMSs Third Biannual Meeting "Summit on Environmental Modelling and Software", Burlington, USA.*

Kralisch, S., P. Krause, M. Fink, C. Fischer, and W.-A. Flügel, (2007), Component based environmental modeling using the JAMS framework. *Proc. of the MODSIM.*

Krause, P. and W. Flügel (2001), J2000 – A modelling system for physically based simulation of hydrological processes in large catchments. *Proc. of the MODSIM*, 1(12):2001.

Vrugt, J., H. Gupta, L. Bastidas, W. Bouten, and S. Sorooshian (2003), Effective and efficient algorithm for multiobjective optimization of hydrologic models. *Water Resources Research, 39(8):1214.*

Yapo, P., H. Gupta, and S. Sorooshian (1998), Multiobjective global optimization for hydrologic models *Journal of Hydrology, 204(1):83–97.*