

Multiobjective tuning of Grid-enabled Earth System Models using a Non-dominated Sorting Genetic Algorithm (NSGA-II)

A. R. Price¹, I. I. Voutchkov¹, G. E. Pound¹, N. R. Edwards²,
T. M. Lenton³, S. J. Cox¹ and the GENIE team⁴

1) School of Engineering Sciences, University of Southampton, Southampton, U.K.

2) Earth Sciences, The Open University, Walton Hall, Milton Keynes, U.K.

3) School of Environmental Sciences, University of East Anglia, Norwich, U.K.

4) <http://www.genie.ac.uk/people>

Abstract

The tuning of parameters in climate models is essential to provide reliable long-term forecasts of Earth system behaviour. In this paper we present the first application of the multiobjective non-dominated sorting genetic algorithm (NSGA-II) to the GENIE-1 Earth System Model (ESM). Twelve model parameters are tuned to improve four objective measures of fitness to observational data. Grid computing and data handling technology is harnessed to perform the concurrent simulations that comprise the generations of the genetic algorithm. Recent advances in the method exploit Response Surface Modelling to provide surrogate models of each objective. This enables more extensive and efficient searching of the design space. We assess the performance of the NSGA-II using surrogates by repeating a tuning exercise that has been performed using a proximal analytical centre plane cutting method and the Ensemble Kalman Filter on the GENIE-1 model. We find that the multiobjective algorithm locates Pareto-optimal solutions which are of comparable quality to those obtained using the single objective optimisation methods.

1. Introduction

To make confident predictions about future climate change we must understand past Earth system behaviour. As yet, we do not fully understand the processes and feedbacks that have driven the most fundamental changes in the climate of the past million years; the transitions between glacial and interglacial planetary states. The GENIEfy [7, 16] project has developed a component framework for the construction, execution and management of Earth System Models capable of simulation over millen-

nial timescales. Constituent models of the Earth system (ocean, atmosphere, land surface, biogeochemistry, sea-ice, icesheets, etc.) at varying resolutions, complexity and dimensionality can be coupled together to form a hierarchy of climate models. The focus of the project is to compose models of intermediate complexity that are capable of simulating the Earth system on millennial timescales. A principle aim of the project is to investigate the physical processes and system feedbacks of importance in understanding Earth System behaviour. Grid computing technology is a key enabler within the project and is being harnessed to aid in flexible coupling of component models, subsequent execution of the resulting ESMs and sharing and management of the data that they generate. In particular, we exploit a Grid-enabled problem solving environment which provides functionality to enable concurrent execution of multiple instances of GENIEfy models on computational grids.

In order to simulate at the multi-decadal time scale and beyond, climate models rely heavily on parameterisations of physical processes that occur on comparatively small time and spatial scales. A key concern in climate modelling is therefore to find appropriate values for the model parameters so that the model simulates a reasonable climatology. This is of particular importance within the GENIEfy project where component models, that are often developed independently, are coupled together to form new ESMs. In order to produce stable and sensible model output it is almost always necessary to re-tune the parameters of the coupled system. However, as with many design problems, the nonlinear response of a model to its parameters and the often conflicting tuning objectives make this a difficult problem to solve. Model tuning has therefore been the subject of much study within the GENIEfy project [4, 9, 1, 21, 17, 16].

The general problem of optimising a set of design variables (model parameters) in order to improve a number

of possibly conflicting design objectives is typically approached in one of two ways [15]. One can create a single objective measure of design quality by computing a weighted sum of the individual objectives and seek to find the set of variables that minimise or maximise this measure. Alternatively, designs may be sought that are Pareto optimal; designs that are superior when all objective measures are considered but that may be inferior when a subset of those objectives are considered [22]. Tuning of GENIEfy ESMs has mostly involved the former option, optimisation of a single objective function to improve the model’s fit to observational data. Initial studies of the GENIE-1 model (specifically the C-GOLDSTEIN composition) exploited a Latin hypercube exploration of 12-dimensional parameter space [4] to minimise a weighted sum of model-data mismatch over four model fields. An implementation of the Ensemble Kalman Filter (EnKF) [6], an efficient data assimilation technique, has been used to integrate an ensemble of 54 models to tune the model parameters to minimise the same objective measure [9]. Further study by Beltran *et al.* [1] on the same problem has applied the proximal analytic centre cutting plane method (ACCPM) technique, a centre plane cutting method, which calculates cheap estimates of the local gradients in the objective function to inform an iterative reduction of the parameter space until an optimum is found. And finally, a Kriging approach [13] has been applied to model the underlying objective function using curve-fitting techniques to optimise the C-GOLDSTEIN function [21]. All of these methods successfully improve the model fit to data but operate on a single composite objective function.

To compose a single objective function a decision maker must provide weighting factors for the individual targets. The optimal or best choice for these weightings is often not known *a priori* and the optimisation must be performed with a, possibly arbitrarily, chosen set of weightings. While many sophisticated algorithms can be applied to a single objective problem, the weighting factors can be critical in the subsequent performance of the optimisation. Multiobjective methods avoid this issue because they seek to find a Pareto set of non-dominated solutions. This provides the domain expert with “good quality” solutions in the objective space from which to select candidates for further study.

Evolutionary programming and Genetic Algorithms [19] are ideal for multiobjective methods since they maintain a population of solutions which “evolve” over generations of the algorithm. Such methods can be used to capture a number of Pareto optimal solutions. However, each generation requires multiple evaluations of the underlying function. In the case of GENIEfy this means expensive runs of an Earth System Model. For such methods to be practical the function evaluations must be performed concurrently. The Grid computing software exploited within GENIEfy enables us to exploit appropriate Grid computing resource to perform

function evaluations in a timely and reliable fashion.

In this paper, we show for the first time the application of the multiobjective non-dominated sorting genetic algorithm (NSGA-II) [3] applied to the C-GOLDSTEIN model from the GENIEfy framework. We discuss the complexity of the algorithm and introduce recent developments that have been made to improve the efficiency of the technique by using surrogate models for the individual objectives. We show how our Grid-enabled Problem Solving Environment is exploited in conjunction with the OPTIONS [12] design search and optimisation package to successfully tune the ESM to observational data. In Section 2 we present the C-GOLDSTEIN model and the target data that the model is tuned to. The NSGA-II method is discussed in section 3 and the application of surrogate models is detailed. The Grid-enabled problem solving environment is discussed in section 4. In section 5 we present the results of the tuning exercise and compare the method with the other algorithms that have been applied to the problem. The merits of our approach are discussed in section 6 and conclusions are given in section 7.

2. C-GOLDSTEIN

The C-GOLDSTEIN Earth System Model consists of a 3D ocean component coupled to a 2D energy-moisture-balance model (EMBM) and a 2D sea-ice code and has been reconstructed in the GENIE framework. This model has been studied extensively within the project using a variety of optimisation and data assimilation methods mentioned above. These methods have all been applied to C-GOLDSTEIN in order to tune its equilibrium state to observational data.

Twelve principal transport parameters are identified in *Edwards and Marsh* [4] whose values are poorly constrained but which strongly influence the model climatology. The aim of the tuning exercise is to find the optimal values for these free parameters that produce an equilibrium model end-state with the closest fit to equivalent observational data. In order to reach equilibrium the model is integrated from a uniform initial state for at least 2000 model years to reach a steady end-state. On a typical desktop PC (Pentium IV 2.6GHz) an integration over 2000 model years typically requires approximately 40 minutes of CPU time. In the previous studies of this problem the tuning has been achieved by minimising a single objective function composed of a weighted sum of the RMS errors between model fields s_i and equivalent observational data S_i across four physical fields (i) in the model.

$$f(x) = \sqrt{\frac{1}{N} \sum_{i=1}^N \frac{(s_i(x) - S_i)^2}{\hat{\sigma}_i^2}} \quad (1)$$

The four physical data sets to which the model is tuned consist of 3D ocean temperature (OCNTEMP) and salinity fields (OCNSAL) and 2D atmospheric surface air temperature (SURFTEMP) and humidity (SURFHUM) fields. These data sets are an average over approximately five decades (spanning the period 1948–2002) and obtained from Levitus [18] and the National Centers for Environmental Prediction (NCEP Reanalysis data provided by the NOAA/OAR/ESRL PSD, Boulder, Colorado, USA, from their Web site at <http://www.cdc.noaa.gov/>). The observational data is interpolated onto the model grid and is weighted by the variance of the data for each field $\hat{\sigma}_i^2$. This weighting necessarily leads to some bias in the objective function. The multiobjective approach is not impacted by the choice of weighting factors because Pareto optimal solutions are sought. For this problem the single objective function can be split into its four constituent components.

$$f_i(x) = \sqrt{\frac{(s_i(x) - S_i)^2}{\hat{\sigma}_i^2}} \quad (2)$$

3. Multiobjective Tuning

Genetic Algorithms [19] require a population of data points to be evaluated over multiple generations to reach an optimal solution. The expense of these evaluations sets practical limits on the scale of study that can be achieved. For example, the NSGA-II study of the GENIE-2 model presented in [20] was limited to 50 generations of a 100 member population because individual function evaluations (model runs) required ~ 100 minutes of CPU time. Recent developments have therefore sought to reduce the number of calls to the expensive function evaluations by introducing surrogate modelling techniques that are popular in single objective optimisation. The method we present in this paper combines cheap extensive searching enabled by surrogate modelling with the multiobjective NSGA-II algorithm which allows competitive objectives to be optimised.

3.1 Optimization using surrogates

In engineering problems the cost of evaluating a model is probably the biggest obstacle preventing extensive optimization procedures. In the multiobjective world, the cost is multiplied, because there are multiple expensive results to obtain. For example, direct evaluation of a finite element model can take several days, which makes it impossible to try hundreds or thousands of combinations.

In the single objective world, approaches using surrogate models are fairly well established and have proven to successfully mitigate the problem of computational expense (Figure 1). The key idea is to build a computationally cheaper model; one that can be extensively searched instead

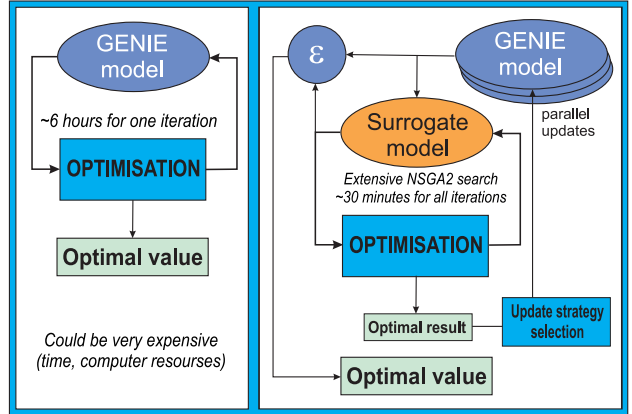


Figure 1. Surrogate models for optimization

of the expensive one. When an optimal solution is found, it needs to be verified on the expensive model and then added to the data pool on which the cheaper model is being trained. The procedure iterates until a good agreement between the two models is achieved or an acceptable solution is found. More and more companies have adopted surrogate assisted optimization techniques and some are making steps to incorporate this approach in their design cycle as a standard. This makes optimization not only useful, but usable and affordable.

There are a number of approximation and curve fitting methods that can be used as surrogates. Polynomial interpolation, cubic splines, neural networks, fuzzy logics and kriging among others all offer different levels of calculation cost vs. accuracy. Choosing an approximation method could be a multiobjective task in its own right. The ideal choice would be the least expensive method that can achieve the best fit to the true model output from Figure 1.

Bearing in mind the complexity of the considered problem, Kriging is the method of choice for this paper. It is well described elsewhere in the literature [14].

3.2 Multiobjective optimization

With advances of computer technology research in multiobjective algorithms quickly bloomed. The ultimate aim now is to produce a set of solutions which, for at least one criterion, are better than the rest. Such a set is called a Pareto front and is known to be the set of non-dominated solutions in the objective space. Scientists could then discuss each of these solutions, compare between requirements and select the best trade-off. This allows several goals to be considered simultaneously and actively searched, aiming to obtain as many solutions as possible, evenly distributed and widely spread around the objective space. Multiobjective algorithms can be classified according to these three criteria, which define the quality of the Pareto front.

As in the single objective methods, various algorithms perform differently for various objective functions. There are a number of methods published and widely used to do this - MOGA, SPEA, PAES, VEGA, NSGA-II, etc. Some are better than others – generally the most popular in the literature are NSGA-II (Deb) and SPEA2 (Zitzler), because they are found to achieve good results for most problems [2, 26]. This paper will present its results using NSGA-II. It is a well known algorithm based on a GA routine. This paper is using the idea of NSGA-II, with further enhanced GA mutation and crossover techniques to refine the algorithm towards optimal performance [12].

The algorithm is further upgraded combining it with Response Surface Model (RSM) techniques, which have shown improvement in the quality of the final Pareto front, as well as significant decrease in the necessary number of expensive function evaluations. Further details can be obtained in the literature [25]. For the convenience of our readers, we would like to sketch out the Response Surface Model process.

1. Perform an initial space filling sampling of the function using a quasi-random $LP\tau$ sequence [23]. Evaluate $N_{LP\tau}$ direct function evaluations
2. Train the Kriging hyper-parameters [11] to provide the best fit of the surrogate model to the data
3. Run NSGA-II on the Response Surface Model using a population size of N_{pop} and evaluating N_{gen} generations of the algorithm
4. Select N_{update} update points to promote diversity in the population using the following strategies
 - Select N_{rnd} random points in the parameter space to help the algorithm escape local minima
 - Take N_{par} evenly spaced points from the RSM Pareto front to advance towards optimal solutions. Such points can however lead to local optima and that is why it should be used in combination with local optimum escaping strategies
 - Perform a small secondary NSGA-II directly on the objective function and select N_{nsga2} . Again such points aid escape from local optima
 - Select N_{err} points of maximum mean squared error selected from the RSM Pareto front. Such points lie in regions where the RSM has worst fit to the data points
 - Select N_{EI} points of maximum expected improvement selected from the RSM Pareto front. Such points indicate regions in the RSM that are statistically most likely to improve the optimum

5. Evaluate these N_{update} points
6. Add the results to the existing data pool of direct function evaluations
7. If the number of data in the pool is larger than N_{krig} points, choose the best N_{krig} points in terms of closeness to the last Pareto front and separation in objective space. The average Euclidian distance between the point being considered and all points on the last Pareto front is used to make this selection. This ensures that the cost of building the RSMs is kept constant and provides the points from the global data pool likely to give the best representation of the underlying functions
8. Rank the data pool of real function evaluations and extract the Pareto front
9. Repeat N_{iter} times from point 2

It should be noted, that the evaluation of the large number of update points N_{update} in step 5 above is made practical by the utilization of Grid computing.

4. Grid-enabled Problem Solving Environment

A principle aim of the GENIEfy project is to exploit Grid computing technology to ease the construction, execution and management of Earth System Models. The software deployed to meet these needs has been built upon output from the GEODISE project [8], a pilot project from the UK e-Science core programme [10]. The GEODISE project has developed toolboxes [5] to bring Grid functionality into the working environments of Scientists and Engineers. In particular, the toolboxes provide intuitive functions in the Matlab and Jython problem solving environments to execute and manage jobs on the computational grid and functions to archive, query and retrieve data on data grids. The GENIE project has exploited these toolboxes to provide a set of function calls that can be used to execute ESM simulations on Compute and Data grids. These functions collectively comprise the GENIE Toolbox with which models from the framework can be managed.

In addition to the GEODISE toolboxes we also exploit the OPTIONS Design Search and Optimisation package [12]. This software provides a suite of sophisticated algorithms for design search and optimisation. An interface to the OPTIONS system has been developed for Matlab to enable these tools to be exploited in conjunction with the Grid functionality available in the Geodise toolboxes.

To configure the study a user simply writes a submission and retrieval function within the Matlab environment. The submission script accepts as input an array of parameter values which it uses to configure and instantiate a model run on a specified Grid resource. The function returns a

data structure describing the job which the optimiser can use to monitor the progress of the simulation. The retrieval function processes the job upon completion and returns the multiple objective function evaluations for that point.

For the studies presented in this paper each function evaluation performs a 2000 year simulation of C-GOLDSTEIN so that the model reaches equilibrium. The typical run time for such a model run is approximately 40 minutes on a modern desktop. The scripting is straight-forward and exploits a small number of calls to functions in the GENIE toolbox. The user typically edits template scripts and is simply required to modify three workspace variables that describe the local runtime environment, the remote resource on which the model will be executed and the complete configuration of the model instance. The latter data structure is populated with the parameter values passed in to the submission function via the argument list. Within this function it is simple to change the Grid resource on which the simulations will be performed by modifying the resource metadata structure.

For this work we exploit the Condor workload management system for high throughput computing [24]. A particular strength of Condor is its ability to pool idle workstations within an institution and harness what would otherwise be wasted CPU cycles. This is an ideal resource since we typically need to perform a large number (50–400) of relatively short simulations concurrently. Dedicated HPC clusters could be targetted but there is a danger of jobs sitting in the queues of such systems at times of high activity. We target a large Condor pool comprising in excess of 1,100 nodes available at a member institution of the project. For this particular pool the nodes are exclusively running the Windows (Win32) operating system which means that Condor’s checkpointing mechanisms are not available. In order to guarantee reliable throughput we therefore queue each evaluation twice on the Condor pool so that inevitable interruptions on individual nodes cause as little disruption to the tuning process as possible. The pool we are using consists of teaching laboratory machines and job eviction rates vary at different times through the academic year. The duplication of effort is therefore justified and Condor’s user priority mechanisms ensure that other users of the pool receive a fair share of the resource.

Three methods are provided to interface to Condor from the Matlab Problem Solving Environment (see Figure 2). The CondorNative toolbox assumes that the Matlab session is executing on a valid Condor submission node and provides functions to submit jobs from a local operating system shell. If the system running the Matlab session does not have direct access to a Condor pool then we provide two toolboxes to enable the use of remote condor submit nodes. The CondorSSH toolbox enables the Matlab session to use SSH (authenticated with the user’s local credentials) to transfer files to a valid Condor submit node and

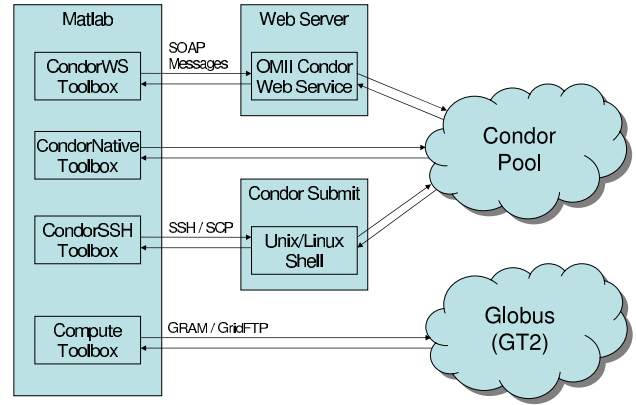


Figure 2. Grid computing infrastructure

invoke jobs from the remote host. If firewall settings prevent the use of SSH then the OMII CondorWS service can be used. This exposes a Condor pool using Web Services and is accessed by the CondorWS toolbox in the Matlab environment. Web services allow a pool to be made accessible beyond the institutional administrative domain by exploiting SOAP messaging which is sent over HTTP. The CondorSSH and CondorWS toolboxes allow the intensive computations required for the Kriging optimisation to be performed on dedicated systems that would otherwise have to double up as submission nodes on a Condor pool.

5. Results

The NSGA-II method with surrogate modelling has been applied to the C-GOLDSTEIN problem. Three independent applications of the method have been executed to investigate its performance and assess the importance of some of its settings. Our first study involved an initial $LP\tau$ sampling of 60 points with 30 update points evaluated at each iteration of the algorithm. A maximum of 120 data points were used to build the Krig metamodels of the four objectives. Once the models were generated the NSGA-II algorithm was applied on the surrogate models of the objectives using a population of 50 members over 50 generations to locate Pareto optimal solutions. The 30 updates to the data set were then selected and evaluated on the Condor pool. The progress of this optimisation is plotted (circles) in the bottom left of Figure 3 as measured by the point with the minimum error value in the single objective space (1). The method was run for 20 iterations and reached its optimal solution after the 17th update.

This initial assessment of the method yielded a result in single objective space that falls between the values obtained using the Proximal-ACCPM and EnKF parameter sets. This in itself is an improvement on the results from Kriging of the single objective reported in [21]. Each iteration of the

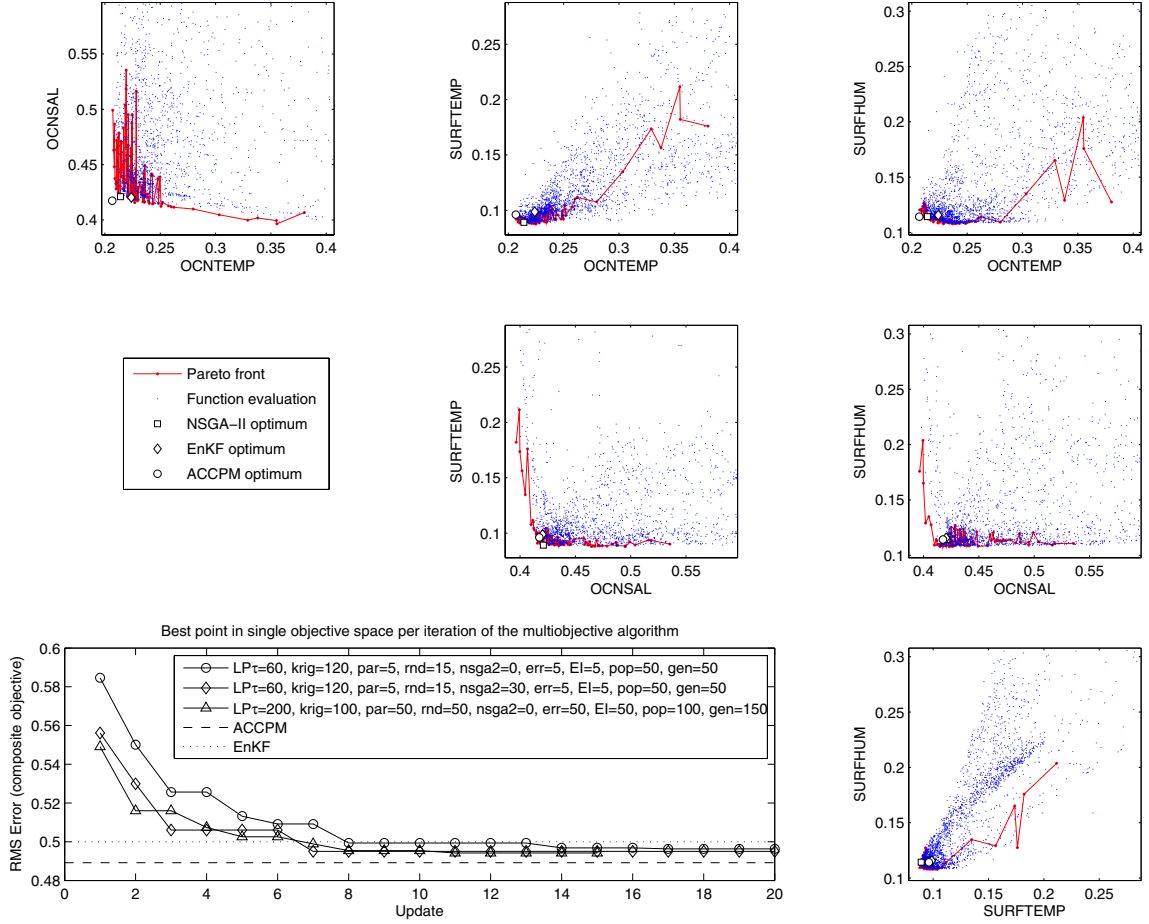


Figure 3. Function evaluations projected onto 2D objective space for each pair of objectives. Bottom left: Progress of the NSGA-II algorithm.

Table 1. Estimated parameters

Parameter	Min	Max	Estimated parameters			Units
			EnKF	Proximal-ACCPM	NSGA-II	
<i>Ocean</i>						
Wind-scale	1.0	3.0	1.6674	1.1841	1.3005	—
Horizontal diff.	3.0×10^2	1.0×10^4	4.1264×10^3	5.5321×10^3	4.0385×10^3	$m^2 s^{-1}$
Vertical diff.	2.0×10^{-6}	2.0×10^{-4}	1.8134×10^{-5}	3.8818×10^{-5}	1.3661×10^{-5}	$m^2 s^{-1}$
Inverse drag	5.0×10^{-1}	5.0	3.4331	4.9959	4.9365	days
<i>Atmosphere</i>						
T diffusion amp	1.0×10^6	1.0×10^7	3.7548×10^6	2.5839×10^6	3.8687×10^6	$m^2 s^{-1}$
Q diffusion amp	5.0×10^4	5.0×10^6	1.7447×10^6	1.9337×10^6	1.4779×10^6	$m^2 s^{-1}$
T advection coeff.	0.0	1.0	6.0357×10^{-2}	8.9163×10^{-2}	5.0122×10^{-2}	—
Q advection coeff.	0.0	1.0	1.3674×10^{-1}	1.4885×10^{-2}	1.4121×10^{-1}	—
FWFlux factor	0.0	2.0	8.9796×10^{-1}	1.0406	8.1917×10^{-1}	(0.32 Sv)
T diffusion width	5.0×10^{-1}	2.0	1.3071	1.9920	1.0528	radians
T diffusion slope	0.0	2.5×10^{-1}	6.8597×10^{-2}	2.3644×10^{-1}	2.3652×10^{-1}	—
<i>Sea-ice</i>						
Sea-ice diffusion	5.0×10^2	8.0×10^3	6.2494×10^3	7.9913×10^3	5.3429×10^3	$m^2 s^{-1}$

algorithm required approximately 2 hours to build the Krig metamodels of the four objectives, roughly 30 minutes to search the models using the NSGA-II algorithm (50 generations) and about an hour to execute the updates.

The addition of 30 independent NSGA-II evaluations at each iteration of the algorithm improves the rate of convergence (diamonds). Our third experiment (triangles) attempted to exploit the surrogate models more extensively. The initial sampling was increased to 200 members of an $LP\tau$ sequence from which a maximum of 100 data points were used to generate the Krig metamodel. The NSGA-II algorithm was applied using a larger population size of 100 and run for 150 generations to perform a more extensive search of the modelled objectives. A total of 200 concurrent model updates were then performed on the Condor pool to improve the data set for the next iteration. From this particular run of the method we obtain our best result in terms of minimum value of the composite objective function (1).

The 2D projections of the function evaluations are plotted for each pair of objectives in Figure 3. These plots give an indication of the scope of the objective space and the nature of the competition between the individual design goals. The Proximal-ACCPM result (circle) lies at the “cusp” of all six plots close to the origin in each case. From our limited sampling, the location of the circle suggests that there exist regions in the parameter space for which each objective can achieve a low value without compromise to one or more of the other objectives. The optimal parameter set obtained using the NSGA-II with surrogates is summarised in Table 1 along with the optimal parameter sets obtained using the EnKF and proximal-ACCPM methods.

6. Discussion

The performance of the NSGA-II with surrogates method is sensitive to its parameters and it would be a tuning exercise in its own right to find the optimal settings for the algorithm in the context of the C-GOLDSTEIN problem. We have performed three independent runs with different parameter settings to get an initial feel for the applicability of the method. In the context of the C-GOLDSTEIN model evaluated over 2000 model years the method appears to work very well. Each application of the method yields a result in between the values of the function evaluated for the parameters obtained from the EnKF and Proximal-ACCPM. It would appear (from our extremely limited sampling) that in the C-GOLDSTEIN model it is possible to find regions of the parameter space for which all objectives are close to their minimum achievable values. For this problem the single composite objective function is therefore a useful tool for selecting points from the Pareto front.

The NSGA-II with surrogate modelling requires the evaluation of approximately an order of magnitude more

model integration years than the Proximal-ACCPM and about double that of the EnKF. There is also additional cost in building the Krig metamodels at each iteration of the algorithm. While the method is more computationally expensive than the single objective methods it provides the project with the ability to find Pareto-optimal solutions. In the context of problems where the individual targets are highly competitive the results from a single objective function will be highly dependent on the weighting factors in the summation. Seeking Pareto optimal solutions opens the door to solving these more difficult problems.

The NSGA-II could be run without surrogate models. In practice we would be limited to a population size of 200 (400 Condor jobs) but could perform a generation of the algorithm every hour. A total of approximately 50 generations could be realistically evaluated every 2 days and clearly the algorithm could be left to run indefinitely. However, using surrogate modelling, an equivalent NSGA-II search over 50 generations can be performed in approximately 30 minutes. Through judicious updates the metamodels are refined and the extensive searching performed on an improved representation of the underlying functions. A comparison of the NSGA-II with and without surrogate modelling will be one focus of future work.

The OPTIONS optimisation package and Geodise Grid-software is deployable at any GENIEfy member institution. With appropriate authentication users can interface to the UK National Grid Service (<http://www.ngs.ac.uk>) or local resource managed by the Globus Toolkit or Condor. As described above the user simply wraps their instance of GENIE model by exposing it as a function in the Matlab environment accepting as input the tuneable parameters and returning, after simulation, the multiple objective measures of fitness to data. The “end-to-end” time for performing a model tuning exercise using this method is therefore relatively low and does not necessitate expert knowledge in the method. The optimiser is presented with a single data structure describing the problem (variables, upper and lower limits, NSGA-II parameters, etc.) and can be invoked by the user with a single call. The time taken from a “decision to study” to an end result is in practice no more than a week. In this regard the method is competitive with the EnKF and Proximal-ACCPM which require expert knowledge to use.

7. Conclusions

Multiobjective optimisation methods avoid the need for formulating a single composite objective function when studying an optimisation problem with competitive design goals. This paper has presented the first application of the multiobjective NSGA-II method using surrogate models to a model from the GENIEfy Earth system modelling framework. Twelve parameters of the GENIE-1 model (C-

GOLDSTEIN) have been tuned to improve its fit to observational climate data. Grid computing has been exploited to perform concurrent runs of the underlying function to provide updates to the data set from which the metamodels are generated. The method obtained a Pareto optimal set of solutions which were compared to solutions obtained using a single composite objective with the Ensemble Kalman Filter and Proximal ACCPM methods. The best solution found using the NSGA-II method is comparable to the best points from the alternative methods.

Acknowledgements The GENIE and GENIEfy projects are funded by the Natural Environment Research Council (NER/T/S/2002/00217 & NE/C515904) through the e-Science programme. Development of the OptionsNSGA2 software has been sponsored by Rolls Royce under the VIVACE project.

References

- [1] C. Beltran, N. R. Edwards, A. Haurie, J.-P. Vial, and D. S. Zachary. Oracle-based optimisation applied to climate model calibration. *Environmental Modelling and Assessment*, 11(1):31–43, 2006.
- [2] K. Deb. *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, Ltd., New York, 2003.
- [3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
- [4] N. R. Edwards and R. Marsh. Uncertainties due to transport-parameter sensitivity in an efficient 3-D ocean-climate model. *Climate Dynamics*, 24:415–433, 2005.
- [5] M. H. Eres, G. E. Pound, Z. Jiao, J. L. Wason, F. Xu, A. J. Keane, and S. J. Cox. Implementation and utilisation of a Grid-enabled problem solving environment in Matlab. *Future Generation Computer Systems*, 21(6):920–929, 2005.
- [6] G. Evensen. The ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53(4):343–367, 2003.
- [7] The GENIE project. <http://www.genie.ac.uk>.
- [8] The GEODISE project. <http://www.geodise.org>.
- [9] J. C. Hargreaves, J. D. Annan, N. R. Edwards, and R. Marsh. An efficient climate forecasting method using an intermediate complexity Earth System Model and the ensemble Kalman filter. *Climate Dynamics*, 23:745–760, 2004.
- [10] T. Hey and A. E. Trefethen. The UK e-science core programme and the Grid. *Future Generation Computing Systems*, 18(8):1017–1031, 2002.
- [11] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.
- [12] A. J. Keane. *The OPTIONS design exploration system: reference manual and user guide*. <http://www.soton.ac.uk/~ajk/options.ps>, 2003.
- [13] A. J. Keane. Wing optimization using design of experiment, response surface, and data fusion methods. *Journal of Aircraft*, 40(4):741–750, 2003.
- [14] A. J. Keane. Statistical improvement criteria for use in multiobjective design optimization. *AIAA Journal*, 44(4):879–891, 2006.
- [15] A. J. Keane and P. B. Nair. *Computational Approaches for Aerospace Design*. Wiley, Chichester, England, U.K., 2005.
- [16] T. M. Lenton, Y. Aksenov, J. D. Annan, T. Cooper-Chadwick, S. J. Cox, N. R. Edwards, S. Goswami, J. C. Hargreaves, P. P. Harris, Z. Jiao, V. N. Livina, D. J. Lunt, R. Marsh, A. J. Payne, A. R. Price, A. J. Ridgwell, I. C. Rutt, J. G. Shepherd, P. J. Valdes, G. Williams, M. S. Williamson, and A. Yool. A modular, scalable, Grid ENabled Integrated Earth system modeling framework: Effect of dynamical atmosphere model, surface grid, and ocean resolution on the stability of the thermohaline circulation. Submitted to *Climate Dynamics*, 2006.
- [17] T. M. Lenton, M. S. Williamson, N. R. Edwards, R. Marsh, A. R. Price, A. J. Ridgwell, J. G. Shepherd, S. J. Cox, and T. G. team. Millennial timescale carbon cycle and climate change in an efficient Earth system model. *Climate Dynamics*, 26(7–8):687–711, 2006.
- [18] S. Levitus, T. P. Boyer, M. E. Conkright, T. O’Brien, J. Antonov, C. Stephens, L. Stathoplos, D. Johnson, and R. Gelfeld. World ocean database 1998 volume 1: Introduction. In *NOAA Atlas NESDIS 18*. U.S. Government Printing Office, Washington, D.C., 1998.
- [19] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. AI Series. Springer-Verlag, New York, 1992.
- [20] A. R. Price, Z. Jiao, I. I. Voutchkov, T. M. Lenton, G. Williams, D. J. Lunt, R. Marsh, P. J. Valdes, S. J. Cox, and the GENIE team. Collaborative study of GENIEfy Earth system models using scripted database workflows in a Grid-enabled PSE. In S. J. Cox, editor, *Proceedings of the UK e-Science All Hands Meeting 2006*, pages 574–581, Nottingham, UK, 2006. EPSRC.
- [21] A. R. Price, G. Xue, A. Yool, D. J. Lunt, P. J. Valdes, T. M. Lenton, J. L. Wason, G. E. Pound, S. J. Cox, and the GENIE team. Optimisation of integrated earth system model components using Grid-enabled data management and computation. *Concurrency and Computation: Practice and Experience*, in press, 2006.
- [22] N. Srinivas and K. Deb. Multiobjective function optimization using nondominated sorting genetic algorithms. *Evol. Comp. J.*, 2(3):221–248, 1995.
- [23] R. B. Statnikov and J. B. Matuzov. *Multicriteria Optimization and Engineering*. Chapman and Hall, New York, 1995.
- [24] D. Thain, T. Tannenbaum, and M. Livny. Distributed computing in practice: the Condor experience. *Concurrency and Computation: Practice and Experience*, 17(2–4):323–356, 2005.
- [25] I. I. Voutchkov and A. J. Keane. Multiobjective optimization using surrogates. In *Proc. 7th Int. Conf. Adaptive Computing in Design and Manufacture*, pages 167–175, Bristol, 2006.
- [26] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computational journal*, 8(2):125–148, 2000.