Efficient global optimization of nonconvex and nondifferentiable climate control problems

A. S. Lieber[1], C. M. Moles[2], J. R. Banga[2], and K. Keller[3,*]

[1] Please fill in

[2] Please fill in

[3,*] Corresponding author:
Princeton Environmental Institute, Princeton University,
Princeton, N.J., 08544, klkeller@princeton.edu, Fax: (609) 258-1274

Abstract

We explore numerical solution techniques for nonconvex and nondifferentiable economic optimal growth models. As an illustrative example, we consider the optimal control problem of choosing the optimal greenhouse gas emissions abatement to avoid or delay a abrupt and irreversible climate damages. We analyze several stochastic global optimization methods such as controlled random search, multi-level coordinate search or differential evolution. The different evolution (DE) algorithm gave the best results in terms of objective function and execution speed. Other techniques are faster than DE in obtaining a local optimum close to the global optimum but misconverge ultimately. The solution time for DE rises approximately exponentially with increasing problem dimension. The nonconvex problem in DE requires up to 50to converge than the convex problem. A simple parallelism scheme without sub-populations decreases the overall solution time for low (around 4) degree of parallelization.

Introduction

Choosing an intelligent policy towards potential impacts of climate change might be seen as an optimal control problem. In other words, present and future societies might want to find some optimal balance between the costs of reducing greenhouse gas emissions and the benefits of avoided climate damages. This kind of integrated economic and scientific assessment has been pioneered by the groundbreaking studies of William Nordhaus (e.g., *Nordhaus* [1992]) and has spawned the field of integrated assessment (IA) of climate change.

The response of the climate-economy system in the integrated assessment models is typically described as smooth, continuous and convex (e.g., *Nordhaus* [1994], or *Nordhaus and Boyer* [2000]). Optimizing a smooth and convex system is relatively straightforward with the help local optimization packages such as GAMS/MINOS. However, the climate-economy system is not smooth and shows significant hysteresis responses [*Rahmstorf*, 1997; *Broecker*, 2000; *Keller et al.*, 2001] which introduce local optimal optima into the economic model. The local optima can render the analysis of climate policy a difficult global optimization problem as they preclude the use of the standard optimization methods. Here we analyze the performance of recent global optimization algorithms with respect to their ability to identify the global optimum and the time required to obtain the best solution.

Integrated assessment models of climate change face three main optimization challenges. First, the time scale of the problem is very long (several centuries) relative to the time scale of economic change. To solve this relatively stiff system requires a large number of time grid-points and results in a large number of optimization variables. Second, the possibility of abrupt climate change introduces jumps in the objective function which choke optimization algorithms relying on partial derivatives. Third, some climatic changes show hysteresis effects which render the models nonconvex (i.e., they introduce local maxima).

The typically applied local optimization techniques find local maxima extremely fast, but often misconverge in nonconvex functions. To avoid misconvergence requires global optimization (GO) techniques — that is algorithms that identify the best (global) optimum and without being trapped in local optima. Computational GO techniques can be broadly broken into two categories-stochastic and deterministic. Stochastic algorithms cannot be guaranteed to converge to absolute suprema during any particular run, but seek to bias random processes towards asymptotic convergence. Deterministic algorithms can be shown to converge to suprema with certainty, but their executions' required time scales, computational resources, and knowledge of the objective function's form may be formidable. Therefore, optimizing detailed, numerically intensive integrated assessment models of climate systems creates challenges even for current state of the art algorithms.

This paper presents both an examination of the performance of a variety of GO techniques and a computational "cookbook" method for optimizing a class of such problems. Our investigation evaluates how different algorithms confront a global optimization task. We analyze the quality of found supremum (function value and vector form), the time in arriving at that answer, the convergence path (how soon the found answer gets close to the best), the relationship between the dimensionality of the problem and solution time, as well as implementation issues on multiprocessing workstations and supercomputer clusters.

To introduce the class of problems examined, section two contains a brief explanation of threshold modified DICE and presents the reasons for its behavior. Section three is an exposition of the behavior of several GO algorithms with consideration for their strengths and weaknesses. Finally, section four explores the performance of Differential Evolution [*Storn and Prince*, 1997] in convex and nonconvex settings as well as an extension in the parallel execution.

The optimization problem

The optimization problem presented in this paper is based on the DICE (Dynamic Integrated model of Climate and the Economy) model, which is a widely used dynamic economic model of the climate change. It integrates economics, carbon cycles, climate science and impacts allowing the weighing of the costs and benefits of taking steps to slow greenhouse warming.

The original formulation of the model held climate damages to economies to be a smooth quadratic relation to change in global mean temperature. *Keller et al.* [2001] modifies the DICE model [*Nordhaus*, 1994] by considering the economic damages caused by an ocean circulation change (technically known as a North Atlantic Thermohaline circulation, THC, collapse). Once crossed, the threshold is irreversible. The damages wrought by crossing the threshold are abrupt and can be severe. Those attributes join to give the system many local extrema and discontinuities. Figure 1 demonstrates the effects of the threshold.

For this paper, we analyze the effects of a climate threshold, that is, a potential ocean thermohaline circulation collapse. The main objective is the maximization of social well-being with a particular set of decisions about investment and CO_2 abatement over time. Although results are reported for the period from year 1995 until 2155, a longer time horizon of 470 years is used to avoid end effects. This results in 94 decision variables (47 for abatement and 47 for investment). This paper considers THC specific damages of 0% and 1.5% of GWP. For summary of the mechanics of the DICE model, see Appendix A.

Methods

Global optimization methods can be roughly classified as deterministic [*Grossmann*, 1996; *Pinter*, 1996; *Horst and Tuy*, 1996] and stochastic strategies ([*Ali et al.*, 1997; *Torn et al.*, 1999]. It should be noted that, although deterministic

methods can guarantee global optimality for certain GO problems, no algorithm can solve general GO problems with certainty in finite time [*Guus et al.*, 1995]. In fact, although several classes of deterministic methods (e.g. branch and bound) have sound theoretical convergence properties, the associated computational effort increases very rapidly (often exponentially) with the problem size.

In contrast, many stochastic methods can locate the vicinity of global solutions with relative rapidity, but with the caveat that global optimality cannot be guaranteed. In practice, stochastic methods often provide the user with a very good (often, the best available) solution in modest computation time. Creating additional speed, many stochastic methods lend themselves to easy parallelization, which increases the size of the problem that can be handled in reasonable wall clock time. Because the problem at hand may be treated as a black box, stochastic methods are usually quite simple to implement. Those characteristics are especially interesting since the researcher often must link the optimizer with a third-party software package where the process dynamic model has been implemented.

Overview of Go methods

We consider a set stochastic GO methods which can handle black box models. The algorithms are included based on their published performance and on our own experiences. Although none of these methods can guarantee optimality, the researcher may solve a given problem with a number of different methods and make a decision based on the set of solutions found. Usually, several of the methods will converge to essentially the same solution. Such multiple convergence should not be regarded as a confirmation of global optimality (it might be the same local optimum), but it does give the user some confidence. Furthermore, it is usually possible to have estimates of lower bounds for the black box cost function and its different terms, so the goodness of the 'global' solution can be evaluated. Sometimes a 'good enough' solution is sufficient.

The considered GO methods are:

ICRS: a stochastic GO method presented by *Banga and Casares* [1987], improving the Controlled Random Search (CRS) method of *Goulcher and Casares* [1978]. Basically, ICRS is a sequential (one trial vector at a time), adaptive random search method which can handle inequality constraints via penalty functions.

LJ: another simple stochastic algorithm, described by *Luus and Jaakola* [1973]. LJ considers a population of trial vectors at each iteration. Constraints are also handled via penalty functions.

DE: the Differential Evolution method, as presented by *Storn and Prince* [1997]. DE is a heuristic, population-based approach to GO. The original code of the DE algorithm [*Storn and Prince*, 1997] does not check if the new generated vectors are within their bound constraints, so we slightly modify the code for that purpose. A brief description of DE is contained in Appendix B.

GCLSOLVE: a deterministic GO method, implemented in Matlab as part of the optimization environment TOMLAB [*Holmstrom*, 1999]. It is a version of the direct algorithm [*Holmstrom*, 1999] that handles nonlinear and integer constraints. GCLSOLVE runs for a predefined number of function evaluations and considers the best function value found as the global optimum.

MCS: the Multilevel Coordinate Search algorithm by *Huyer and Neumaier* [1999], inspired by the DIRECT method [*Jones*, 2001], is an intermediate between purely heuristic methods and those allowing an assessment of the quality of the minimum obtained. It has an initial global phase after which a local procedure, based on a SQP algorithm, is launched. These local enhancements lead to quick convergence if the

global phase has found a point in the basin of attraction of the global minimizer.

GLOBAL: this is a hybrid GO method by *Csendes* [1988], which essentially is a modification of the algorithm by *Boender et al.* [1982]. This method uses a random search followed by a local search routine. Initially, it carries out a clustering phase where the Single Linkage method is used. Next, two different local search procedures can be selected for a second step. The first (LOCAL) is an algorithm of Quasi-Newton type that uses the DFP (Davison-Fletcher-Powell) update formula. The second, more appropriate for problems with discontinuous objective functions or derivatives, is a robust random search method (UNIRANDI) by *Jarvi* [1973].

Adopted Methods

To attack the integrated climate control problem of interest, we present and compare the solutions obtained to the problem using several global optimization (GO) methods. Each of the GO methods considered has several adjustable search parameters which can greatly influence their performance, both in terms of efficiency and robustness. We have followed the published recommendations for each method (see references cited above) together with the feedback obtained after a few preliminary runs. All the computation times reported here were obtained using a low cost platform, PC Pentium III/866 MHz. For the sake of a fair comparison, we consider Matlab implementations of all these methods, except for the case of GLOBAL, where only a FORTRAN implementation, difficult to translate to Matlab, was available.

We conducted tests for the case of THC specific damages of 1.5% of GWP (Gross World Product) (case named theta15). In order to illustrate the comparative performance of multi-start local methods for this type of problem, a

multi-start code (named ms-FMINU) is also implemented in Matlab making use of the FMINU code, which is a part of the MATLAB Optimization Toolbox (1994)[*Grace*, 1994]. FMINU is meant for unconstrained functions. Its default algorithm is a quasi-Newton method that uses the BFGS formula for updating the approximation for the Hessian matrix. Its default line search algorithm is a safe-guarded mixed quadratic and cubic polynomial interpolation and extrapolation method.

RESULTS AND DISCUSSION

When solving a global optimization problem, it is usually interesting to estimate how multi-modal it is. In order to illustrate its non-convexity, the problem is solved using the multi-start (ms-FMINU) approach, considering 100 random initial vectors generated satisfying the bounds of the decision variables. This strategy converges to a large number of local solutions, as depicted in the histogram shown in figure 2. It is very significant that despite the huge computational effort associated with the 100 runs, the best value found (C*=23584.71) is still far from the solutions the GO methods obtain with much smaller computation times. These results illustrate the inability of the multi-start approach to handle highly multi-modal problems like this one.

For the theta15 problem considered here, the best value of C* = 26398.8 is obtained by DE (in the C implementation) after several restarts and very long iterations [*Keller et al.*, 2001]. In this study, the Matlab implementation of DE finds again the best result (see Table 1). It converges to essentially the same result as in *Keller et al.* [2001], C*=26398.71, although the associated computational effort is relatively large. The MCS method also converges to a quite good result (C*=26397.00) but in less than 250 seconds, i.e. 1.5 orders of magnitude faster than DE. The ICRS method is able to arrive at a reasonably good value in just 10 minutes of computation. Table 1 presents the best objective value each method obtains, plus the corresponding CPU times and the required number of function

evaluations. No results are presented regarding the GLOBAL algorithm, which does not converge successfully, probably due to the relatively large dimensionality of the problem.

Comparing methods based only final objective function values and the associated computation times neglects their intermediate behavior. In order to provide a more complete comparison of the different methods, a plot of the convergence curves (objective function values represented as relative error versus computation time) is presented in figure 3, where curves for each method are plotted. It can be seen that the ICRS method presented the most rapid convergence initially, but is ultimately surpassed by DE. It should be noted that, although most of the computation time is employed in function evaluations, the methods differed regarding the computational overhead needed for the generation of new trial vectors. Stochastic methods generate new search directions using simple operations, so they have the minimum associated overhead. The reverse happens with deterministic methods.

Additionally, the solutions from ICRS and LJ, DE from random start vectors are compared in figure 4. Depending on the starting conditions, ICRS and LJ can get trapped in local optima. DE, for the generated vectors, always arrives at very close to the best found solution. To extend the comparison of solutions, the solution vector forms for the various GO methods are compared with that of Keller by inspecting the decision variables values at the different optima. Plots of the decision variables for the different solutions are presented in figure 5. It should be noted that there are significant differences in the optimal abatement policies, although the objective function values are quite similar. This indicates a very low sensitivity of the cost function with respect to that control variable, a rather frequent result in dynamic optimization problem.

DE Performance

Having established that DE outperforms all other considered algorithms, we now explore the performance characteristics of an implementation of DE. First, we explore the convergence of DE on one CPU under different dimensionalities of problems. The theta15 case is used as in the preceeding GO survey. But, the number of optimization variables is varied from ten to eighty in increments of ten. It is observed how many function evaluations and how much wall-clock time was necessary to converge close to the final value. Values are recorded for achieving 99.9%, 99.99%, and 99.999% of eventually converged solution. The model was executed with THC specific damages of 1.5% GWP and with 0% GWP. The wall clock results are captured in the figure below.

Performance tests in this section are all performed on a Beowulf cluster. It consists of 67 networked workstations. Each workstation has 128 megabytes of RAM and two 350MHz Intel Pentium II processors. They are connected using high speed non-blocking switches such that communications latency is a fraction of a millisecond.

Figure 6 presents the results for different numbers of dimensions. Time requirements rise exponentially as the number of dimensions to optimize increases. To arrive at the time required for partial convergence, three runs starting from high, medium, and low initial conditions (all 0.99, 0.50, and 0.01 decision variables respectively) are used and their times are averaged. Time required for the presence and absence of THC when only 99.9% convergence is required lie nearly colinear. In all other cases, the presence of positive THC specific damages increases the time required for a specific level of convergence. The difference in convergence times for different models increases as the required level of accuracy increases. That is, for positive levels of THC specific damages, it is more difficult to increase accuracy than for runs with zero THC specific damages.

To examine another aspect of DE's behavior, an experimental implementation

of parallel executing DE was tested. Trial vectors are generated on a master node just as in the original DE code, but then instead of locally evaluating the objective function, the trial vectors are passed to slave nodes on the network. Those slave nodes received the trial vectors and the resulting values are returned. The performance gains from such a regime rely critically on the ratio of the amount of work involved in an objective function evaluation and the communication delays involved in parsing out the work to slave nodes. Figure 7 captures the results for this test.

The need for fast communications is especially true in cluster implementations. Runs with learning and parameter uncertainty require far more computation per candidate vector than runs without. Moving from one to two CPUs yields a roughly 35% performance boost. This is because both CPUs reside in the same computer and communications are performed entirely in memory. When moving from two to four processors, network communication is involved and the performance boost is only a further 10%. This is because the same population size is used as in earlier runs. Bandwidth is high, but latency is significant. With successively larger numbers of CPUs working on the same number of function evaluations, each one has less work to do, but nearly the same amount of communications overhead. Therefore, under this scheme, adding larger numbers of CPUs has diminishing productivity. Perhaps in the future a researcher will implement a fully distributed DE algorithm where candidate vectors are generated and evaluated on slave nodes. Considerable experimentation will be required to find good heuristic crossover rates between nodes.

Conclusion

We analyze a variety of optimization techniques for a nonconvex and nondifferentiable optimal control problem arising from the analysis of climate change policy. The Differential Evolution algorithm achieves the best solution as measured both

by the value of the objective function and the smoothness of policy recommendation.

Furthermore, we evaluate the performance parameters of the DE algorithm. Under conditions of convexity and nonconvexity, different magnitudes of dimensionality are explored for their impact on computation time. As the dimensionality of the problem increases, the time required for DE convergence rises roughly exponentially. Convex problems require less solution time than nonconvex problems. A simple parallelization scheme improves the solution time for small numbers of CPUs.

Appendix A: DICE Model

The DICE model is a long term dynamic model of optimal economical growth that links economic activities and climate change. The model uses an increasing transform of the discounted flow of utility across time to measure well-being and rank economic activities and possibilities and climate outcomes and consequences. Well-being is represented in the model by a flow of utility U to society, defined as the product of the logarithm of per capita consumption per year c, and the exogenously given population L:

$$U(t) = L(t)\ \ln\ c(t). \tag{1}$$

Nordhaus defined utility as income, and we depart from that in order to place value on equity and reduced variability.

To differentiate between present and future utility, a "pure rate of social time preference," ρ, is applied to discount the future. Recognizing that abatement costs apply to the present as well as future, but benefits lie only in the future, the choice model seeks to maximize the discounted sum of well-being expressed as (U^*):

$$U^* \ = \ \sum_{t=t_o}^{t^*} U(t)\ (1+\rho)^{-t}, \tag{2}$$

which is calculated by discounting the flow of utility, not income, at time t from some starting point t_o to an appropriate time horizon t^*.

The gross world product (GWP, Q) is to be determined by a Cobb-Douglas production function of capital K and population with the parameters: level of technology A, output scaling factor Ω, and elasticity of output γ with respect to capital:

$$Q(t) = \Omega(t)\ A(t)\ K(t)^{\gamma}\ L(t)^{1-\gamma}. \tag{3}$$

GWP takes into account abatement costs and climate damages, but not capital depreciation. The effect of abatement costs and climate related damages on output is incorporated into the model via the output scaling factor (see equation 16).

Feasible consumption paths depend on the economy's output. Total consumption C is the difference between gross world product and gross investment I:

$$C(t) = Q(t) - I(t). \tag{4}$$

To link economic activity to causes of environmental effects, the DICE model assumed that carbon emissions, E, during one year into the atmosphere are proportional to the gross world product, with the proportionality determined by the exogenous carbon intensity of production σ and the policy choice of the level of carbon emissions abatement μ:

$$E(t) = [1 - \mu(t)]\, \sigma(t)\, Q(t). \tag{5}$$

A constant fraction β of carbon emissions is added to the atmospheric carbon stock M (the rest is assumed to be absorbed by carbon sinks). A portion δ_M of the atmospheric carbon in excess of the preindustrial stock of 590 Gt is diffused during each time step to the deep ocean so that the atmospheric stock evolves according to:

$$M(t) = 590 + \beta\, E(t-1) + (1 - \delta_M)[M(t-1) - 590]. \tag{6}$$

Atmospheric carbon dioxide acts as a greenhouse gas, causing a change F in the radiative forcing from the preindustrial level according to:

$$F(t) = 4.1\, \frac{\ln(M(t)/590)}{\ln(2)} + O(t), \tag{7}$$

where O represents the (exogenously determined) change in forcing due to other greenhouse gases like methane or CFCs. An increase in radiative forcing causes an increase in global mean atmospheric temperature T from its preindustrial level, which is modeled using a simple atmosphere-ocean climate model according to:

$$\begin{aligned} T(t) = T(t-1) + (1/R_1)[F(t) - \lambda T(t-1) \\ -(R_2/\tau_{12})(T(t-1) - T^*(t-1))]. \end{aligned} \tag{8}$$

In this equation R_1 and R_2 denote the thermal capacity of the oceanic mixed layer and the deep ocean, respectively, λ is the climate feedback parameter, τ_{12} is the transfer rate from the oceanic mixed layer to the deep ocean, and T^* is the deviation of the deep-ocean temperature from the preindustrial level approximated by:

$$T^*(t) = T^*(t-1) + (1/R_2)[(R_2/\tau_{12})(T(t-1) - T^*(t-1))]. \tag{9}$$

A key property of the climate system is the "climate sensitivity," which is the hypothetical increase in equilibrium temperature for a doubling of atmospheric CO_2, placed by the IPCC between 1.5 and 4.5 degrees Celsius per doubling of CO_2. In the DICE model, the climate sensitivity is inversely related to the parameter λ. Specifically, the modeled climate sensitivity is given by the ratio of the increase in radiative forcing for a doubling of atmospheric CO_2 (equal to 4.1, equation 21) to λ.

The damages relative to gross world product (D) are assumed to be a function of the deviation of the global average temperature from it's preindustrial value:

$$D(t) = \theta_1 \, T(t)^{\theta_2}, \tag{10}$$

where θ_1 and θ_2 are model parameters. The cost of CO_2 emissions abatement TC, measured as a fraction of gross world product, is given by:

$$TC(t) = b_1 \, \mu(t)^{b_2}, \tag{11}$$

where b_1 and b_2 are model parameters. Given the calculated abatement costs and climate damages, global output is re-scaled with the scaling factor Ω:

$$\Omega(t) = [1 - TC(t)]/(1 + D(t)). \tag{12}$$

This scaling factor approximates the effects of small damages reasonably well, compared to the explicit accounting, which would imply $\Omega(t) = 1 - TC(t) - D(t)$.

Model parameter values are used from the original DICE model, with one exception. We adopt a climate sensitivity of 3.6 degrees Celsius per doubling of CO_2 instead of the previously used 2.9 degrees as our standard value. Based on the analysis of climate data and the expert opinion of the IPCC, Tol98 estimate the values of the median and the standard deviation of the climate sensitivity as 3.6 and 1.1 ^{o}C.

We depart from the original DICE model structure in two ways: (i) We consider damages caused by an uncertain environmental threshold imposed by an ocean circulation change (known as a North Atlantic Thermohaline circulation (THC) collapse); (ii) we examine uncertainties by posing the model as a probabilistic optimization problem.

Ocean modeling studies suggest that the THC may collapse when the equivalent CO_2 concentration $P_{CO_{2,e}}$, the concentration of CO_2 and all other greenhouse gases expressed as the concentration of CO_2 that leads to the same radiative forcing) rises above a critical value ($P_{CO_{2,e,crit}}$) (e.g., [*Stocker and Schmittner*, 1997]). We represent this phenomenon in the model by imposing a threshold specific climate damage (θ_3) for all times after the THC has collapsed. We approximate $P_{CO_{2,e}}$ by an exponential fit to previously calculated stabilization levels in the DICE model [*Keller et al.*, 2000]. It is important to note that some of our calculations extrapolate beyond the 2 to 4 ^{o}C range of climate sensitivity explored by Stocker and Schmittner (1997).

Second, we explore the effects of parameter uncertainty on a policy that maximizes the expected value of the objective function. To this end, we formulate the model as a probabilistic optimization problem. Because we use a numerical solution method (discussion follows), we consider only a discrete subsample ("states of the world" [SOW]) from the continuous probability density functions. We maximize the expected value of the total discounted utility U^* over all states of the world, weighted by their probability. Expected utility maximization is used as a decision criterion to be consistent with previous studies [*Nordhaus*, 1994;

Nordhaus and Popp, 1997].

Appendix B: DE Algorithm

Differential Evolution [DE] is an algorithm first described by Storn and Price in 1995. It is an approach to global optimization suited to nonlinear and non differentiable spaces where direct search approaches are best suited. A relatively new algorithm, DE is not as commonly used as Simulated Annealing (SAs) and Genetic Algorithms (GAs). Unlike GAs, but like SAs, DEs perform mathematically meaningful operations on its vectors, not their binary stored values. At its heart, DE is a suite of methods for combining and evaluating candidate vectors in a feasible space. This section describes its operation.

First, NP parameter vectors are chosen at random for form the initial population. Generation size does not vary across time. If nothing is known of the objective functions behavior, the initial population members may be distributed uniformly. Otherwise, they might be distributed normally around a trial solution. Trial parameter vectors for the next generation are generated by adding the weighted difference vector between two population members to a third member. If the resulting vector yields a lower objective function value than a predetermined population member, the newly generated vector replaces the vector with which it was compared. The best parameter vector $x_{best,G}$ is evaluated for every generation G in order to track minimization progress. The method (DE2) used in this paper for generating trial vectors is described below.

For each vector $x_{i,G}, i = 0, 1, 2, ..., NP - 1$, a trial vector v is generated according to the following rule.

$$v = x_{i,G} + \lambda * (x_{best,G} - x_{i,G}) + F * (x_{r_2,G} - x_{r_3,G}) \tag{13}$$

The control variable λ controls the greediness of the scheme by determining how heavily to weight the current best vector $x_{best,G}$. To increase variation of the parameter vectors, the vector $u = (u_1, u_2, ..., u_D)^T$ with

$$u = v_j \; for \; j =< n >_D, ... < n + L - 1 >_D \; otherwise = (x_{i,G})_j \tag{14}$$

where $<>_D$ denote the modulo function with modulus D. In other words, some dimensions of the vector u acquire the values of v, while others equal the original values of $x_{i,G}$. This is similar to crossover in Genetic Algorithms. The integer L is the crossover probability and, along with n, is a random decision for each trial vector v. If the objective function at u is an improved state compared to $x_{i,G}$, u is retained. The reverse is also true.

References

Ali, M., C. Storey, and A. Torn, Application of stochastic global algorithms to practical problems, *Journal of Optimal Theory Applications, 95*, 545, 1997.

Banga, J. R., and J. J. Casares, Integrated controlled random search: application to a wastewater treatment plant model, *IchemE Symp. Series, 100*, 183, 1987.

Boender, C., A. H. G. Rinnooy Kan, and G. T. Timmer, Stochastic method for global optimization, *Math Programming, 22*, 125, 1982.

Broecker, W. S., Abrupt climate change: causal constraints provided by the paleoclimate record, *Earth-Science Reviews, 51*(1-4), 137–154, 2000.

Csendes, T., Nonlinear parameter estimation by global optimization - efficiency and reliability, *Acta Cybernetica, 8*, 361, 1988.

Goulcher, R., and J. J. Casares, The solution of steady-state chemical engineering optimization problems using random search technique, *Computer Chemical Eng., 2*, 33, 1978.

Grace, A., Optimization toolbox user's guide, *The Math Works Inc.*, 1994.

Grossmann, I. E., Global optimization in engineering design, *Kluwer Academic Publishers*, 1996.

Guus, C., C. Boender, and E. Rineuhbm G, Stochastic methods, *Handbook of Global Optimization*, 1995.

Holmstrom, J., The tomlab optimization environment in matlab, *Advanced Model. Optimization, 1*(1), 47, 1999.

Horst, R., and H. Tuy, Global optimization - determiniztic approaches, *Springer -Verlag*, 1996.

Huyer, W., and A. Neumaier, Global optimization by multilevel coordinate search, *Journal of Global Optimization, 14*, 331–335, 1999.

Jarvi, T., A random search optimiser with an application to a maxmin problem, *Publications of the Institue of Applied Mathematics, 3*, 1973.

Jones, D. R., Direct, *In encyclopedia of Optimization*, 2001.

Keller, K., B. M. Bolker, and D. F. Bradford, Uncertain climate thresholds and economic optimal growth, *Workshop on Potential Catastrophic Impacts of Climate Change*, 2001.

Keller, K., K. Tan, F. M. M. Morel, and D. F. Bradford, Preserving the ocean circulation: Implications for climate policy, *Climatic Change, 47*, 17–43, 2000.

Luus, R., and T. H. I. Jaakola, Optimization by direct search and systematic reducation of the size of search region, *AIChE, 19*, 760, 1973.

Nordhaus, W. D., An optimal transition path for controlling greenhouse gases, *Science, 258*, 1315–1319, 1992.

Nordhaus, W. D., *Managing the global commons: The economics of climate change*, The MIT press, Cambridge, Massachusetts, 1994.

Nordhaus, W. D., and J. Boyer, *Warming the World: Economic Models of Global Warming*, MIT Press, 2000.

Nordhaus, W. D., and D. Popp, What is the value of scientific knowledge? an application to global warming using the price model, *The Energy Journal, 18*(1), 1997.

Pinter, J. D., Global optimization in action, *Kluwer Academic Publishers*, 1996.

Rahmstorf, S., Risk of sea-change in the Atlantic, *Nature, 388*, 825–826, 1997.

Stocker, T. F., and A. Schmittner, Influence of co2 emission rates of the stability of the thermohaline circulation, *Nature, 388*, 862–865, 1997.

Storn, R., and K. Prince, Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces, *Journal of Global Optimization, 11*, 341, 1997.

Torn, A., M. Ali, and S. Viitanen, Stochastic global optimization: problem classes and solution techniques, *Journal of Global Optimization, 14*, 437, 1999.

Appendix A: DICE Model

The DICE model is a long term dynamic model of optimal economical growth that links economic activities and climate change. The model uses an increasing transform of the discounted flow of utility across time to measure well-being and rank economic activities and possibilities and climate outcomes and consequences. Well-being is represented in the model by a flow of utility U to society, defined as the product of the logarithm of per capita consumption per year c, and the exogenously given population L:

$$U(t) = L(t)\ \ln\ c(t). \tag{15}$$

Nordhaus defined utility as income, and we depart from that in order to place value on equity and reduced variability.

To differentiate between present and future utility, a "pure rate of social time preference," ρ, is applied to discount the future. Recognizing that abatement costs apply to the present as well as future, but benefits lie only in the future, the choice model seeks to maximize the discounted sum of well-being expressed as (U^*):

$$U^* \ = \ \sum_{t=t_o}^{t^*} U(t)\ (1+\rho)^{-t}, \tag{16}$$

which is calculated by discounting the flow of utility, not income, at time t from some starting point t_o to an appropriate time horizon t^*.

The gross world product (GWP, Q) is to be determined by a Cobb-Douglas production function of capital K and population with the parameters: level of technology A, output scaling factor Ω, and elasticity of output γ with respect to capital:

$$Q(t) = \Omega(t)\ A(t)\ K(t)^{\gamma}\ L(t)^{1-\gamma}. \tag{17}$$

GWP takes into account abatement costs and climate damages, but not capital depreciation. The effect of abatement costs and climate related damages on output is incorporated into the model via the output scaling factor (see equation 16).

Feasible consumption paths depend on the economy's output. Total consumption C is the difference between gross world product and gross investment I:

$$C(t) = Q(t) - I(t). \tag{18}$$

To link economic activity to causes of environmental effects, the DICE model assumed that carbon emissions, E, during one year into the atmosphere are proportional to the gross world product, with the proportionality determined by the exogenous carbon intensity of production σ and the policy choice of the level of carbon emissions abatement μ:

$$E(t) = [1 - \mu(t)]\, \sigma(t)\, Q(t). \tag{19}$$

A constant fraction β of carbon emissions is added to the atmospheric carbon stock M (the rest is assumed to be absorbed by carbon sinks). A portion δ_M of the atmospheric carbon in excess of the preindustrial stock of 590 Gt is diffused during each time step to the deep ocean so that the atmospheric stock evolves according to:

$$M(t) = 590 + \beta\, E(t-1) + (1 - \delta_M)[M(t-1) - 590]. \tag{20}$$

Atmospheric carbon dioxide acts as a greenhouse gas, causing a change F in the radiative forcing from the preindustrial level according to:

$$F(t) = 4.1\, \frac{\ln(M(t)/590)}{\ln(2)} + O(t), \tag{21}$$

where O represents the (exogenously determined) change in forcing due to other greenhouse gases like methane or CFCs. An increase in radiative forcing causes an increase in global mean atmospheric temperature T from its preindustrial level, which is modeled using a simple atmosphere-ocean climate model according to:

$$\begin{aligned} T(t) = T(t-1) + (1/R_1)[F(t) - \lambda T(t-1) \\ -(R_2/\tau_{12})(T(t-1) - T^*(t-1))]. \end{aligned} \tag{22}$$

In this equation R_1 and R_2 denote the thermal capacity of the oceanic mixed layer and the deep ocean, respectively, λ is the climate feedback parameter, τ_{12} is the transfer rate from the oceanic mixed layer to the deep ocean, and T^* is the deviation of the deep-ocean temperature from the preindustrial level approximated by:

$$T^*(t) = T^*(t-1) + (1/R_2)[(R_2/\tau_{12})(T(t-1) - T^*(t-1))]. \tag{23}$$

A key property of the climate system is the "climate sensitivity," which is the hypothetical increase in equilibrium temperature for a doubling of atmospheric CO_2, placed by the IPCC between 1.5 and 4.5 degrees Celsius per doubling of CO_2. In the DICE model, the climate sensitivity is inversely related to the parameter λ. Specifically, the modeled climate sensitivity is given by the ratio of the increase in radiative forcing for a doubling of atmospheric CO_2 (equal to 4.1, equation 21) to λ.

The damages relative to gross world product (D) are assumed to be a function of the deviation of the global average temperature from it's preindustrial value:

$$D(t) = \theta_1 \, T(t)^{\theta_2}, \tag{24}$$

where θ_1 and θ_2 are model parameters. The cost of CO_2 emissions abatement TC, measured as a fraction of gross world product, is given by:

$$TC(t) = b_1 \, \mu(t)^{b_2}, \tag{25}$$

where b_1 and b_2 are model parameters. Given the calculated abatement costs and climate damages, global output is re-scaled with the scaling factor Ω:

$$\Omega(t) = [1 - TC(t)]/(1 + D(t)). \tag{26}$$

This scaling factor approximates the effects of small damages reasonably well, compared to the explicit accounting, which would imply $\Omega(t) = 1 - TC(t) - D(t)$.

Model parameter values are used from the original DICE model, with one exception. We adopt a climate sensitivity of 3.6 degrees Celsius per doubling of CO_2 instead of the previously used 2.9 degrees as our standard value. Based on the analysis of climate data and the expert opinion of the IPCC, Tol98 estimate the values of the median and the standard deviation of the climate sensitivity as 3.6 and 1.1 ^{o}C.

We depart from the original DICE model structure in two ways: (i) We consider damages caused by an uncertain environmental threshold imposed by an ocean circulation change (known as a North Atlantic Thermohaline circulation (THC) collapse); (ii) we examine uncertainties by posing the model as a probabilistic optimization problem.

Ocean modeling studies suggest that the THC may collapse when the equivalent CO_2 concentration $P_{CO_{2,e}}$, the concentration of CO_2 and all other greenhouse gases expressed as the concentration of CO_2 that leads to the same radiative forcing) rises above a critical value ($P_{CO_{2,e,crit}}$) (e.g., [*Stocker and Schmittner*, 1997]). We represent this phenomenon in the model by imposing a threshold specific climate damage (θ_3) for all times after the THC has collapsed. We approximate $P_{CO_{2,e}}$ by an exponential fit to previously calculated stabilization levels in the DICE model [*Keller et al.*, 2000]. It is important to note that some of our calculations extrapolate beyond the 2 to 4 ^{o}C range of climate sensitivity explored by Stocker and Schmittner (1997).

Second, we explore the effects of parameter uncertainty on a policy that maximizes the expected value of the objective function. To this end, we formulate the model as a probabilistic optimization problem. Because we use a numerical solution method (discussion follows), we consider only a discrete subsample ("states of the world" [SOW]) from the continuous probability density functions. We maximize the expected value of the total discounted utility U^* over all states of the world, weighted by their probability. Expected utility maximization is used as a decision criterion to be consistent with previous studies [*Nordhaus*, 1994;

Nordhaus and Popp, 1997].

Appendix B: DE Algorithm

Differential Evolution [DE] is an algorithm first described by Storn and Price in 1995. It is an approach to global optimization suited to nonlinear and non differentiable spaces where direct search approaches are best suited. A relatively new algorithm, DE is not as commonly used as Simulated Annealing (SAs) and Genetic Algorithms (GAs). Unlike GAs, but like SAs, DEs perform mathematically meaningful operations on its vectors, not their binary stored values. At its heart, DE is a suite of methods for combining and evaluating candidate vectors in a feasible space. This section describes its operation.

First, NP parameter vectors are chosen at random for form the initial population. Generation size does not vary across time. If nothing is known of the objective functions behavior, the initial population members may be distributed uniformly. Otherwise, they might be distributed normally around a trial solution. Trial parameter vectors for the next generation are generated by adding the weighted difference vector between two population members to a third member. If the resulting vector yields a lower objective function value than a predetermined population member, the newly generated vector replaces the vector with which it was compared. The best parameter vector $x_{best,G}$ is evaluated for every generation G in order to track minimization progress. The method (DE2) used in this paper for generating trial vectors is described below.

For each vector $x_{i,G}, i = 0, 1, 2, ..., NP - 1$, a trial vector v is generated according to the following rule.

$$v = x_{i,G} + \lambda * (x_{best,G} - x_{i,G}) + F * (x_{r_2,G} - x_{r_3,G}) \tag{27}$$

The control variable λ controls the greediness of the scheme by determining how heavily to weight the current best vector $x_{best,G}$. To increase variation of the parameter vectors, the vector $u = (u_1, u_2, ..., u_D)^T$ with

$$u = v_j \; for \; j =< n >_D, ... < n + L - 1 >_D \; otherwise = (x_{i,G})_j \tag{28}$$

where $<>_D$ denote the modulo function with modulus D. In other words, some dimensions of the vector u acquire the values of v, while others equal the original values of $x_{i,G}$. This is similar to crossover in Genetic Algorithms. The integer L is the crossover probability and, along with n, is a random decision for each trial vector v. If the objective function at u is an improved state compared to $x_{i,G}$, u is retained. The reverse is also true.

Results	DE	ICRS	LJ	MCS	GCLSOLVE
Neval	$3.5 * 10^6$	386860	20701	71934	65000
CPU time, s	6652.76	600.54	57.32	246.29	62272.64
C*	26398.7133	26383.7162	26375.8383	26397.0090	26377.0649

Table 1: Convergence performance for various GO algorithms. DE takes the most function evaluations to converge, but finds the best solution. Otherwise, MCS gets the closest.

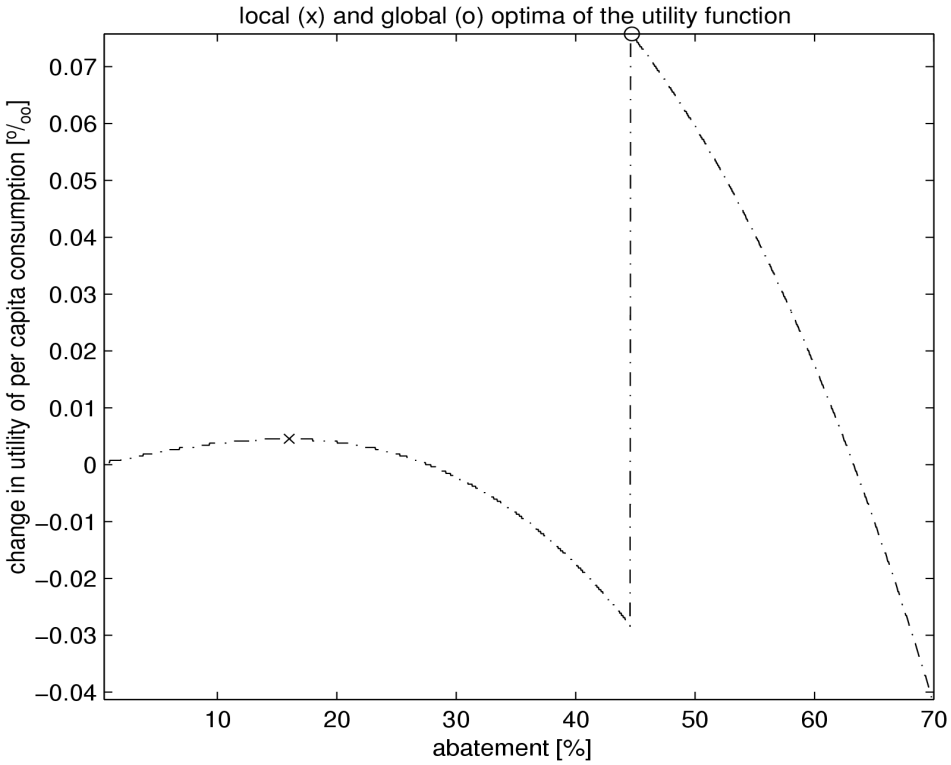

Figure 1: Illustration of how as abatement levels increase for a given period, there can be discontinuities and multiple extrema in utility. From 0% to 45%, the THC is allowed to collapse, and abatement simply alters other climate damages. At 45%, the THC does not collapse and there is an abrupt increase in utility. Beyond 45%, further abatement is yields no further increases in utility in this model.

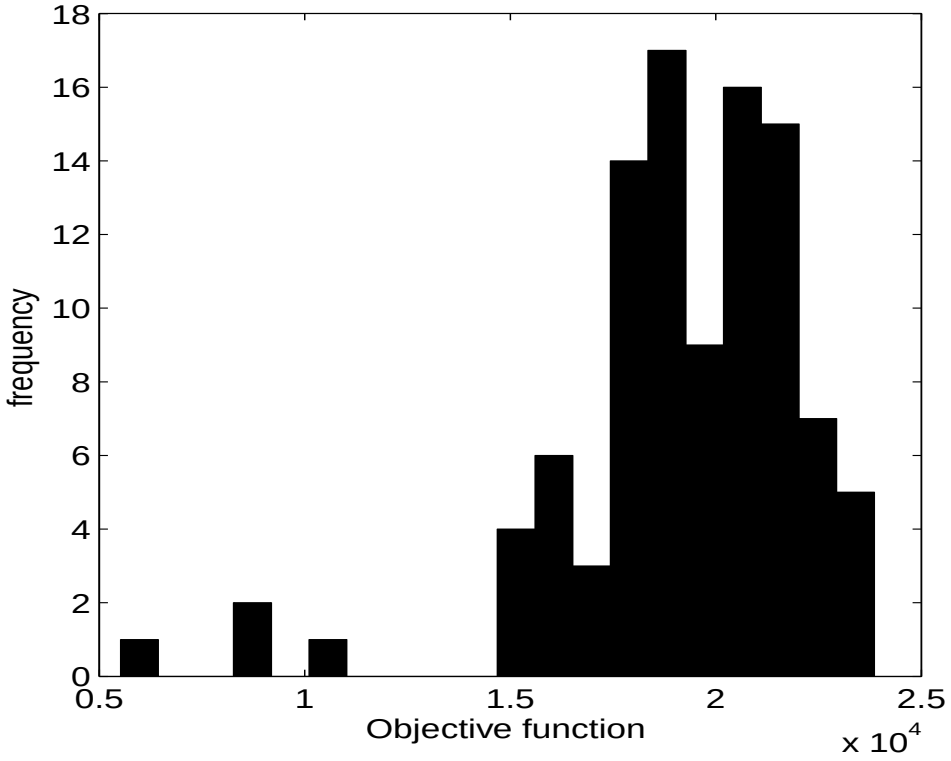

Figure 2: Histogram of results found using multi-start strategy. This demonstrates that local search methods are prone to fail in nonconvex problems. In this multi-modal model, there are many widely scattered local optima, and the search is trapped in local basins of attraction.

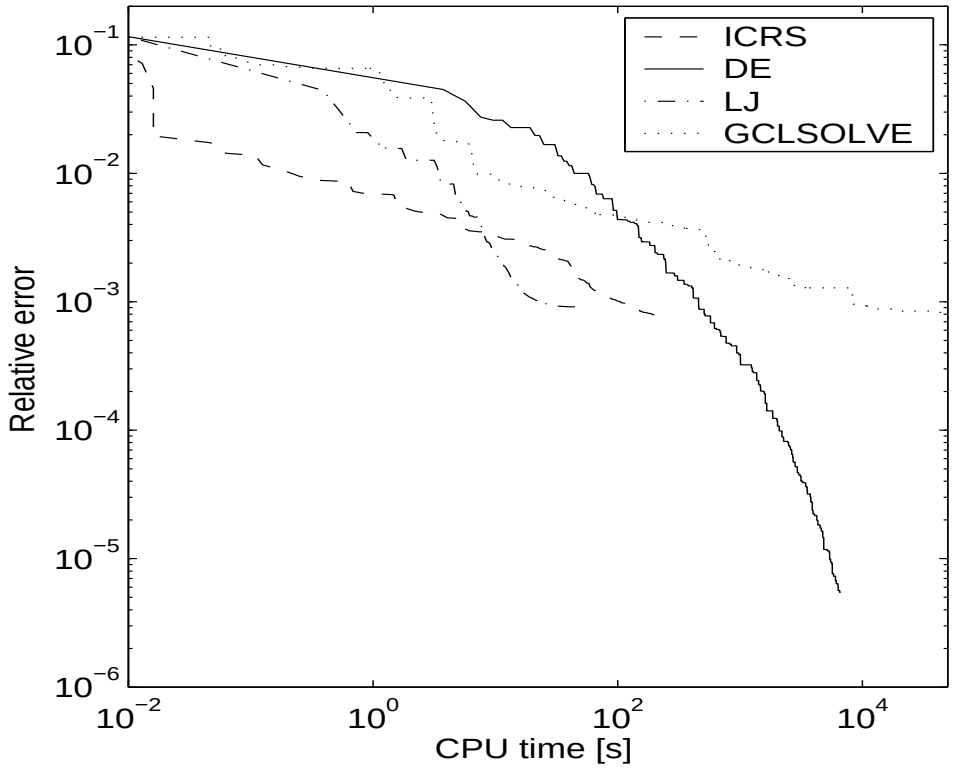

Figure 3: Representation of algorithm convergence paths. DE achieves the best solution, but takes the longest time to get there. ICRS achieves a 'good' solution rapidly, but never improves as far as DE.

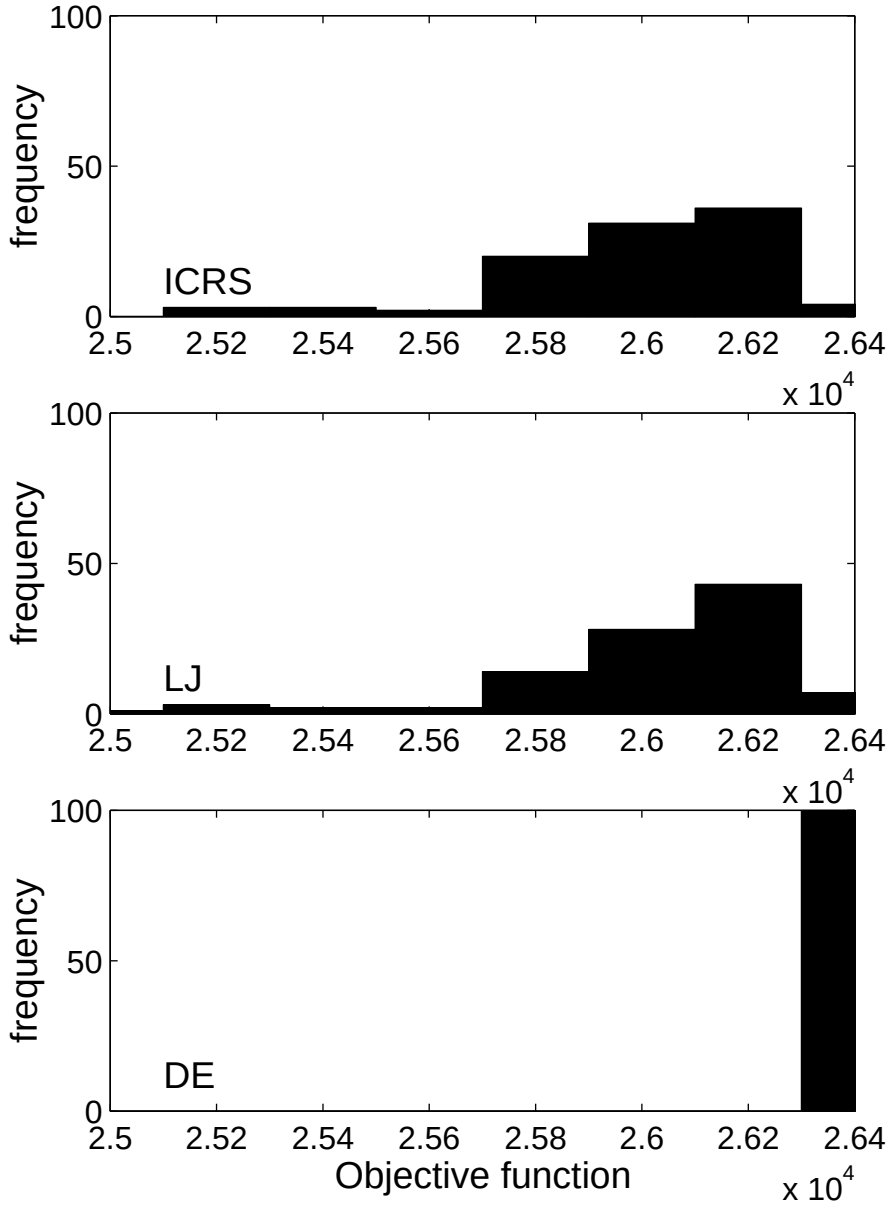

Figure 4: Even GO algorithms can be prone to misconvergence. Here DE, LJ, and ICRS algorithms are restarted from random starting points. DE converges most nearly to the same answer each time, while the others are not as robust.

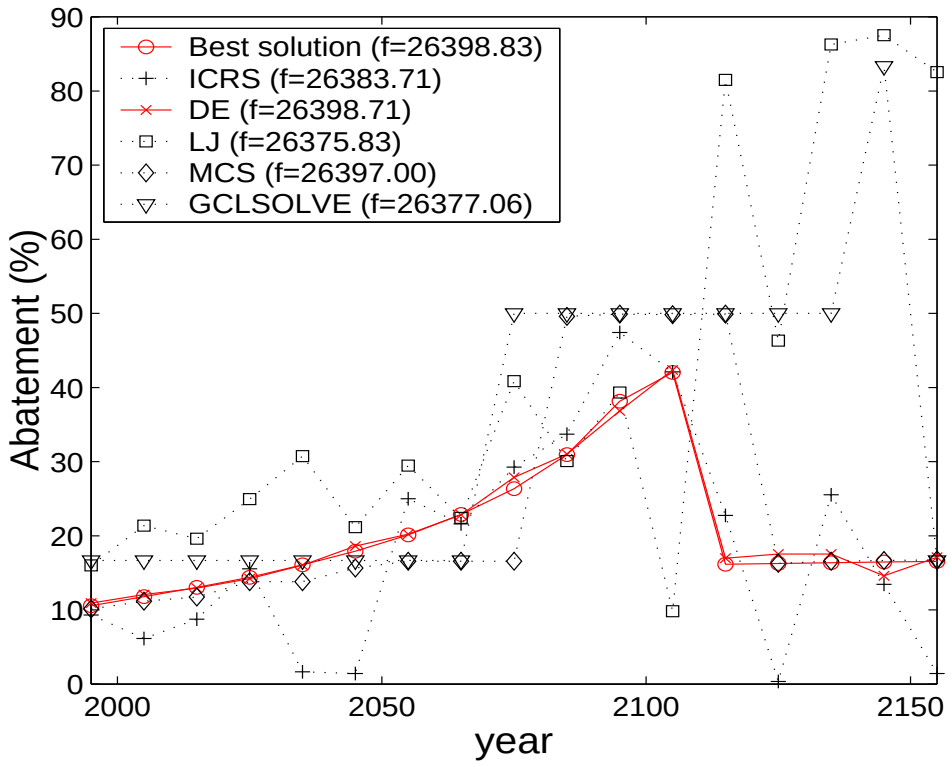

Figure 5: Vector forms of best found solution for various GO algorithms. Many algorithms converge to nearly the same value for the objective function, but the form of the solution widely differs. For some policy implementations, such lack of smoothness can be a critical defect.

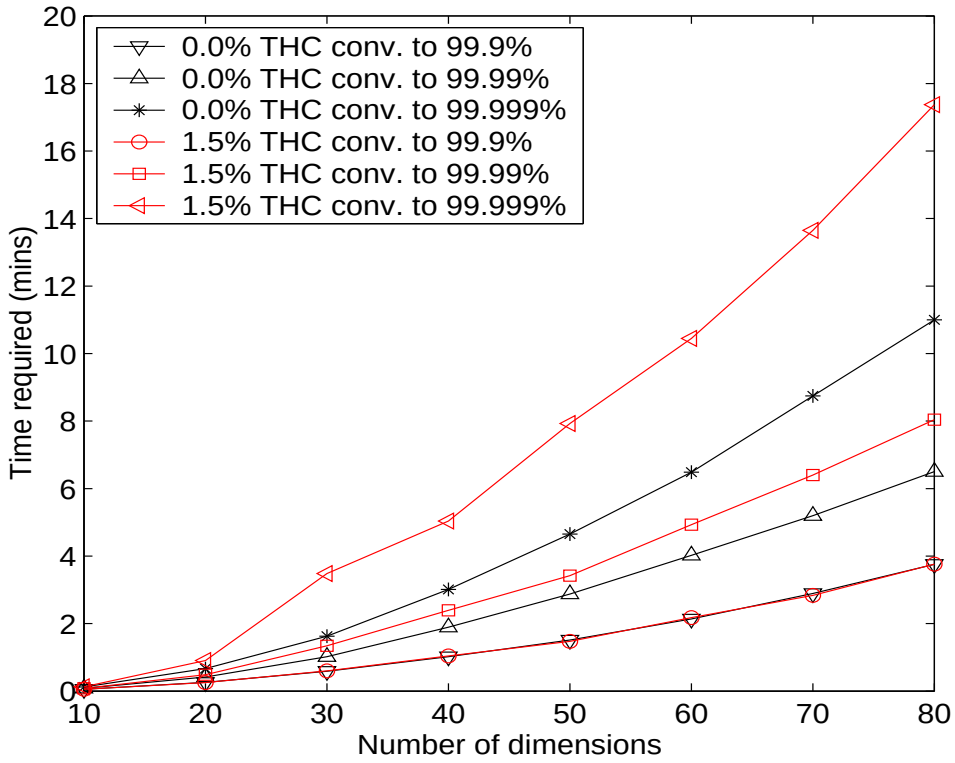

Figure 6: As the number of dimensions in the optimization increases, the time required for convergence by DE increases approximately exponentially. It takes longer to achieve precision in nonconvex problems (where there are positive THC damages), than under convex regimes (no THC damages).

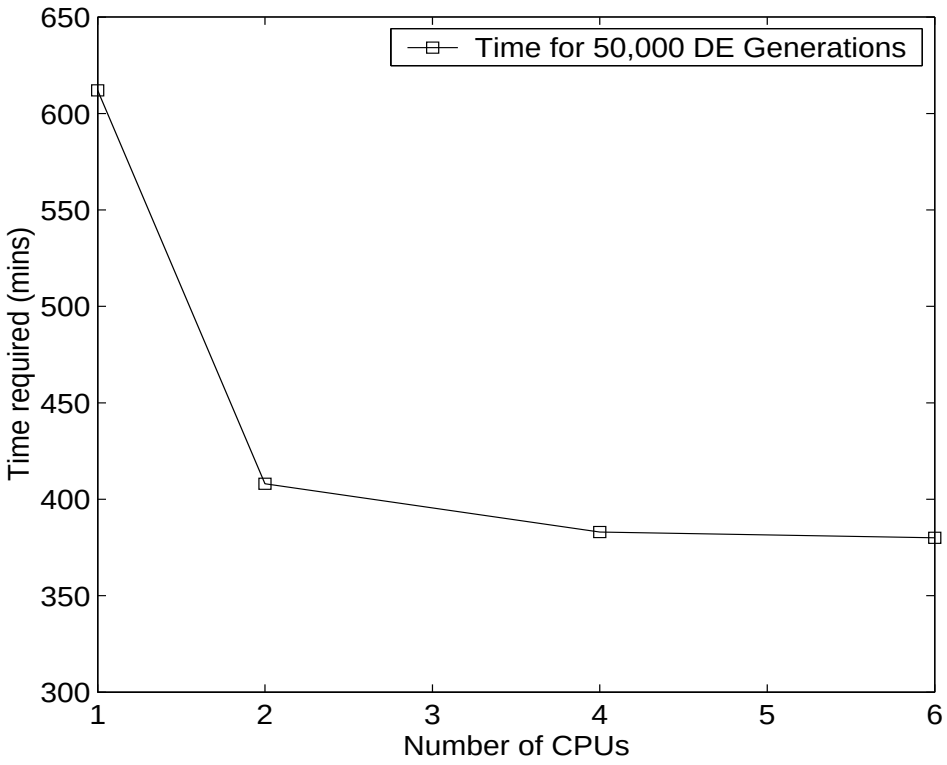

Figure 7: Adding a second processor improves the speed of vector evaluation since in the Beowulf hardware, two CPUs reside in the same machine sharing memory. This provides for rapid communication. Once additional CPUs are added across networks to share the same workload, latency factors in communication diminish additional gains from parallelism.