Deterministic Ensemble Forecasts Using Gene-Expression Programming*

ATOSSA BAKSHAI AND ROLAND STULL

University of British Columbia, Vancouver, British Columbia, Canada

(Manuscript received 31 July 2008, in final form 9 April 2009)

ABSTRACT

A method called gene-expression programming (GEP), which uses symbolic regression to form a nonlinear combination of ensemble NWP forecasts, is introduced. From a population of competing and evolving algorithms (each of which can create a different combination of NWP ensemble members), GEP uses computational natural selection to find the algorithm that maximizes a weather verification fitness function. The resulting best algorithm yields a deterministic ensemble forecast (DEF) that could serve as an alternative to the traditional ensemble average.

Motivated by the difficulty in forecasting montane precipitation, the ability of GEP to produce bias-corrected short-range 24-h-accumulated precipitation DEFs is tested at 24 weather stations in mountainous southwestern Canada. As input to GEP are 11 limited-area ensemble members from three different NWP models at four horizontal grid spacings. The data consist of 198 quality controlled observation–forecast date pairs during the two fall–spring rainy seasons of October 2003–March 2005.

Comparing the verification scores of GEP DEF versus an equally weighted ensemble-average DEF, the GEP DEFs were found to be better for about half of the mountain weather stations tested, while ensemble-average DEFs were better for the remaining stations. Regarding the multimodel multigrid-size “ensemble space” spanned by the ensemble members, a sparse sampling of this space with several carefully chosen ensemble members is found to create a DEF that is almost as good as a DEF using the full 11-member ensemble. The best GEP algorithms are nonunique and irreproducible, yet give consistent results that can be used to good advantage at selected weather stations.

1. Bias-corrected ensembles

An ensemble of outputs from different numerical weather prediction (NWP) runs can provide information on forecast confidence, probability of different forecast outcomes, range of possible errors, predictability of the atmosphere, and other metrics. Although expensive ensemble runs are justified predominantly by their probabilistic output (McCollor and Stull 2009), the most widely used metric is an average of the ensemble members. Except for rare events (Hamill et al. 2000; ECMWF 2008), this ensemble average usually has better skill,

consistency, quality, and economic value than other deterministic NWP forecasts of similar grid resolution (Richardson 2000; Wandishin et al. 2001; Zhu et al. 2002; Kalnay 2003). If we define a deterministic ensemble forecast (DEF) as any combination (linear or nonlinear) of ensemble members, then the ensemble average is one type of DEF.

Raw NWP outputs can have large biases in mountainous regions. Biases can be reduced by performing statistical postprocessing, namely, by regressing previous forecasts against their verifying observations (the training set) and using the resulting regression equation or algorithm to remove biases in future forecasts (Wilks 2006). Postprocessing is an important component of the daily operational runs at NWP centers. A large variety of NWP postprocessing methods have been tested over decades by many investigators. Recent work (Gel 2007; Hacker and Rife 2007; Yussouf and Stensrud 2006, 2007; Yuan et al. 2007; Cheng and Steenburgh 2007; Hansen 2007) includes multiple linear regression/model output statistics (MOS), sequential estimation of systematic error,

* Supplemental information related to this paper is available at the AMS Journals Online Web site: <http://dx.doi.org/10.1175/2009WAF2222192.s1> and <http://dx.doi.org/10.1175/2009WAF2222192.s2>.

Corresponding author address: Roland Stull, Dept. of Earth and Ocean Sciences, University of British Columbia, 6339 Stores Rd., Vancouver, BC V6T 1Z4, Canada.
E-mail: rstull@eos.ubc.ca

updatable MOS, classification and regression trees, alternative conditional expectation, nonparametric regression, running weighted-mean bias-corrected ensemble (BCE), Kalman filtering (KF, a linear approach), neural networks (nonlinear), and fuzzy logic.

DEF (e.g., ensemble averaging) and statistical postprocessing are intertwined. DEF reduces random error, while statistical postprocessing reduces systematic errors (biases). Hence, using both together is recommended to provide an optimum forecast.

An *explicit* way to combine DEF and statistical postprocessing is to first postprocess each ensemble member individually to reduce its bias and, then, average all of the resulting ensemble members. This bias-corrected ensemble (BCE) method separates statistical postprocessing from ensemble averaging. For example, the University of British Columbia (UBC) operational short-range ensemble uses Kalman filtering (Bozic 1994) to remove the biases from individual temperature forecasts before weighting them equally in a “pooled” ensemble average (Delle Monache et al. 2006; McCollor and Stull 2008a). Each station gets its own Kalman filter correction. Other investigators have used running means instead of Kalman filters to perform the bias correction (Stensrud and Skindlov 1996; Stensrud and Yussouf 2003, 2005; Yussouf and Stensrud 2007; Eckel and Mass 2005; Jones et al. 2007; Woodcock and Engel 2005; McCollor and Stull 2008a).

An example of an *implicit* combination is a *weighted* ensemble average, where each ensemble member is weighted inversely to its past forecast error. Here, statistical postprocessing is in the form of the computation of the past error from a training set. Those ensemble members with larger biases will have larger error statistics, giving them smaller weights in the ensemble average—an approach that tends to reduce the overall bias of the ensemble average. Yussouf and Stensrud (2007) found explicit approaches to be slightly better than implicit, while we found (unpublished) nearly identical skill for implicit and explicit equally weighted Kalman filtered (pooled) temperature ensembles at the University of British Columbia (UBC).

Bias-corrected ensemble averages have worked well for variables such as temperature but are more difficult to implement for montane precipitation, perhaps because of its time series with many zeros and its hybrid discrete–continuous frequency distribution [Bernoulli gamma or Poisson gamma; see review by Cannon (2008)]. So motivated, we present a new way of creating bias-corrected DEFs via the gene-expression-programming (GEP) method of evolutionary programming. Section 2 describes the GEP concept. Section 3 explains our motivation and covers case study data and the experimental

procedure. Section 4 shows case study results, and section 5 presents a sensitivity test to determine the most important ensemble members. A discussion of the uniqueness and utility of the algorithms found using GEP is given in section 6, with conclusions in section 7.

2. Evolutionary programming

a. Method

The ensemble average uses three operators (+, ×, /) to combine the individual ensemble members into a DEF. Using 24-h accumulated precipitation (P24) at any one weather station as an example, the algorithm of the simple ensemble average is $\text{DEF}_{\text{P24}} = [c_1 \times \text{P24}_1 + c_2 \times \text{P24}_2 + \dots c_N \times \text{P24}_N]/N$, where N is the number of ensemble members, c are the respective weights, and the subscript indicates the ensemble member.

But suppose we allow more operators, more functions (sine, exponential, if, or, greater than, etc.), and nonlinear as well as linear combinations. An arbitrary example is $\text{DEF}_{\text{P24}} = (\text{P24}_1^{c_1}) \times \sin(\text{P24}_2) + \ln(N)/\exp(c_N \times \text{P24}_N)$. The result is an equation or algorithm to calculate the DEF. Consider this algorithm to be one individual in a population of algorithms. The other individuals are different algorithms using different sets of functions, constants, and operators to estimate the DEF, but each individual uses the same set of N -ensemble P24 inputs and produces a forecast for the same weather station.

If the algorithms are created somewhat randomly, most individuals would give DEFs that do not verify well against observations. We allow those individuals with higher verification scores to spawn slightly modified daughter versions of themselves (with changes to a small portion of their operators, functions, or constants) to create a new population of algorithms. The individuals in this second generation are again culled via computational natural selection for a “training set” of data.

After many generations, the population evolves to favor the stronger individuals, thereby giving us the better algorithms, hence, the name evolutionary programming. The process of finding an algorithm that best reduces some statistical measure of error is called symbolic regression. This regression is run separately at each weather station, yielding individual algorithms with different constants and different functional forms at the different stations. After a best individual algorithm appears in the training dataset, it can be used with the real-time forecasts for that station. As will be shown, the resulting algorithms are often not simple, yet they can give the best bias-corrected DEF at some of the weather stations.

This approach is motivated by the human body. The body is an amazingly complex system of chemical, electrical, hydraulic, pneumatic, and other physical processes

that works exceptionally well. This fact has caused us to reevaluate Occam's razor, one of the philosophical underpinnings of modern science. Instead of believing that the "simplest explanation tends to be the right one," we suggest that "a scientific relationship should not be more complex than needed." Regarding the human body, it is amazingly complex, but not more so than needed.

We suspect that the existing postprocessing and DEF algorithms for most surface weather variables (precipitation, humidity, winds, etc.) are much too simple. The linear approaches of regression and Kalman filtering use only a handful of predictors with roughly an equal number of weights. The math functions they use are only (+, −, ×, and /). Even the neural network assumes a handful of fixed functions (+, −, ×, /, exp, or atan) for the transfer functions between neural nodes and allows only the 70 or so weights to be fit during training.

If we consider the mathematical space of *all* of the possible algorithms, then the current methods of NWP statistical postprocessing and DEF are sampling just a small portion of this space. Evolutionary programming allows us to sample a much broader portion of the algorithm space, and use a much richer set of functions (arithmetic, trigonometric, hyperbolic, relational, etc.) in algorithms that are allowed to grow in complexity as needed to give the best solution.

b. The evolution of evolutionary programming

Evolutionary programming is a type of machine learning–artificial intelligence (Mitchell 1997) designed to perform symbolic regression. Three variants in evolutionary programming are "genetic algorithms," "genetic programming," and "gene-expression programming" (Ferreira 2006). Genetic algorithms (GAs) were devised in the 1960s by Holland (1975), and are coded as symbolic strings of fixed length that work analogous to ribonucleic acid (RNA) replicators in nature. Many of the individuals in a GA population evolve into symbolic strings that produce nonfunctional algorithms, reducing evolutionary efficiency and requiring extra computation to find and remove the nonviable individuals. The "artificial life" (A-life) simulations of Adami (1998) and Lenski et al. (2003) are examples of genetic algorithms, but with a growing computational genome.

Genetic programming (GP) was devised by Cramer (1985) and promoted by Koza (1992, 1994) and Koza et al. (1999, 2003). This method uses nonlinear parse trees of different sizes and shapes, which are analogous to protein replicators in nature. This approach also has the difficulty that many mutations result in nonfunctional algorithms—wasting computations.

GEP is promoted by Ferreira (2001, 2006) as the best evolutionary programming method to date. It mimics a

fully separated genotype (the genetic information) and phenotype (the physical body as informed by the genetic information) system in nature, but has the advantage that it always forms functional individuals (i.e., always produces valid expression trees in computer code). It also allows unconstrained modification of the genome (allowing wider sampling of algorithm space), allows point mutations, and results in extremely rapid computational evolution.

c. Gene-expression programming (GEP)

GEP uses a computational data structure analogous to a biological chromosome to hold the genetic information for an individual algorithm (Ferreira 2006). The GEP chromosome is a string of characters, where some characters represent mathematical operators and others represent operands (called terminals in GEP). The terminals can be the raw output from NWP models (i.e., the predictors) or numbers (analogous to coefficients or weights). Also, the operators can operate on other operators. This chromosome defines a computational expression tree, namely, an individual algorithm. Each character in the chromosome defines a node in the expression tree. A chromosome can be designed to represent a single gene (monogenic) or many genes (multigenic). Each gene gives one expression tree. The multigenic expression trees can be combined via a simple function to make the complete individual. Figure 1 illustrates an algorithm in (a) normal mathematical form, (b) its computational expression tree, and (c) its representation as a monogenic chromosome in GEP.

Different individuals in the initial randomly created population have different chromosomes, defining different expression trees. Each expression tree is evaluated using all the forecast–observation pairs of data in the training set, and a "fitness" of that expression tree is found using a measure of the forecast error.

Fitness can be calculated using any appropriate statistical measure. We utilize mean absolute error (MAE), root relative squared error (RRSE), and relative absolute error (RAE) (see the appendix). We modify the fitness measure to nudge solutions toward parsimony by giving slightly higher fitness to those individuals that are simpler. Individuals that are more fit according to one measure are sometimes less fit according to other measures. We chose an optimum fitness measure by trial and error. Also, as the evolution progresses, we get a more accurate DEF by changing the fitness measure from a statistic that favors outliers (RRSE) to one that does not (MAE).

Individuals that are more fit are given a higher probability of reproducing. This selection is via a computational analogy to roulette wheel sampling (Goldberg

FIG. 1. Different representations of an algorithm, as illustrated using Planck's law. (a) Mathematical representation, where E is the radiant flux, a and b are constants, T is the absolute temperature, and λ is the wavelength. (b) Expression-tree representation, where P raises the first argument to the power of the second, e is the exp function, $*$ is multiplication, $/$ is division, and $-$ is subtraction. All circles are nodes, and the thinner circles are terminal nodes. (c) Chromosome representation in GEP, which can be created by reading the expression tree (b) like a book: from left to right on each horizontal (dashed) line, starting at the top line and working down. Each function or operator has a known arity (number of arguments), which is used when interpreting the chromosome (the genotype) to build the proper connections between nodes in an expression tree (the phenotype). The ellipsis (...) represents additional elements of the chromosome that are not read (i.e., do not contribute to the expression tree).

1989), where each individual receives a sector of the roulette wheel proportional to its fitness. To maintain population size, the roulette wheel is computationally “spun” as many times as there are individuals in the population. Although this method gives a greater chance of selection to the fitter individuals, it also allows for the occasional selection of less-fit individuals, which is important for maintaining genetic diversity in the population.

The combination of input data (forecast–observation pairs), the fitness measure, and the selection mechanism is called the “selection environment.” The selected individuals are then reproduced with chromosome modification to form the next generation. Also, the single best individual in each generation is copied without modification into the next generation.

The modification methods of the GEP chromosome include mutation, inversion, transposition, and recombination (see the supplemental information available at the Journals Online Web site: <http://dx.doi.org/10.1175/2009WAF2222192.s1> and <http://dx.doi.org/10.1175/2009WAF2222192.s2>). The last method is sexual, because it involves the sharing of genetic information be-

tween different individuals within the same generation. Computational evolution is accelerated (relative to biological evolution) by applying these modifications at rates ranging from 0.04 for mutation (e.g., four mutations per 100-character chromosome per generation) to 0.4 for one-point recombination. These computational modifications are patterned after observed biological modifications that are known to have been important in driving the evolution of life (Ferreira 2006).

The process of reproduction, fitness determination, and natural selection is repeated from generation to generation, creating a genetically diverse evolving population. Figure 2 illustrates GEP evolution, showing selected generations that provide successively better fits to a simple noisy dataset.

The population exists in a computational “world” or environment where the individuals can sexually share genetic information during reproduction. But depending on the complexity of the problem, it is possible that none of the individuals in a world will evolve to an acceptable solution. However, multiple worlds (i.e., separate GEP evolutions) can be created on the computer, each with different random initial populations,

<u>Generation</u>	<u>GEP Chromosome</u>	<u>GEP Equation</u>	<u>Simplified Equation</u>
a) 14	/ab...	$y = a / b$	$y = 0.4452$
b) 68	*x//a-+axxx...	$y = x \left[\frac{\left\{ \frac{a-x}{x+x} \right\}}{a} \right]$	$y = 0.5 + 0.119 x$
c) 3967	*x//a-+axx+/x*-+x//xababx...		

$$y = x \left[\frac{\left\{ \frac{a-x}{x + \frac{(x+a)x}{(b/a) - (b/x) + x}} \right\}}{a} \right]$$

$$y = \frac{(x + 3.7856) / 3.7856}{2 + 0.466x \left(\frac{x - 3.7856}{x + 3.7856} \right)}$$

d)

FIG. 2. Examples of a few of the fittest individuals taken from a world containing 50 monogenic GEP individuals as they evolve toward fitting sigmoid artificial data [dots in (d)]. These artificial data were created using an error function $y = 0.5 \cdot [1 + \text{sign}(x) \cdot \text{erf}(|0.5x|)]$ with added random noise uniformly distributed within ± 0.1 . For this illustration, the set of functions available to GEP is limited to $(+, -, *, /)$, there are only two randomly evolving constants (a, b) , the GEP chromosomes each have a head of 20 characters and tail of 21, and fitness is determined using RRSE. Shown for selected generations is the best GEP chromosome within that generation, the equation as literally described by that chromosome, and a manually simplified version of the equation. (a) By generation 14, a reasonable constant y value was born (dashed line), where $(a, b) = (-4.2014, -9.4366)$. (b) By generation 68, a sloping straight line was born (thin solid), where $(a, b) = (-4.2014, -9.4366)$. (c) By generation 3967, a well-fit sigmoid curve was found (thick curved line) with a relative explained variance of $r^2 = 0.979$. The two constants had evolved to $(a, b) = (-3.7856, -8.1229)$. The evolution took 5 min on a dual-core laptop computer.

and each of which could evolve to different end states. New worlds can continue to be created until one yields an acceptable individual (i.e., an algorithm that works well, within some error tolerance).

An important outcome of this evolutionary approach is that different worlds might yield different acceptable

individuals. Namely, there can be an arbitrary number of *nonunique* good solutions, each of which works nearly as well as the others. Figure 3 illustrates this effect, showing nonunique solutions that fit sigmoid data. In statistical postprocessing we are not trying to discover a physical “law,” but instead seek any statistical

FIG. 3. Examples of nonunique fits by GEP to the synthetic sigmoid data (dots) from Fig. 2. Fitness is measured by the relative explained variance, r^2 . (a) When the GEP set of allowable functions is limited to $(+, -, L)$ where L is the logistic function $L(x) = [1 + \exp(-x)]^{-1}$, then GEP evolves to a fittest individual (thick line) of the form $y = L[-0.38745 + x + L(x)]$, with fitness $r^2 = 0.978$. (b) If GEP functions are limited to $(+, -, \text{power})$, then the fittest individual (thin dashed line, mostly hidden behind the thick line) is $y = 0.762055^{[2.388183\{[0.881012^x] - x\}]}$, with $r^2 = 0.978$. (c) If the GEP functions are limited to $(+, -, \text{mod})$ where “mod” is the floating-point remainder (as defined in the Fortran 95/2003 standard, not as defined in Microsoft Excel), then the fittest individual (thin solid line) is $y = 0.5057 + \text{mod}\{\text{mod}\{\text{mod}[x, (-0.468079 - x)], (-0.468079 + x)\}, x\}$, with $r^2 = 0.952$. These last two examples show how evolutionary computation inexorably approaches a best fit, even when it is restricted to an unusual set of functions that a human would not have intuitively used to fit a sigmoid shape.

relationship that works. Uniqueness issues are discussed in section 6.

The details of GEP are described by Ferreira (2006). She uses a more sophisticated approach than illustrated above, with a multigenic chromosome that finds successful solutions more quickly. Ferreira implemented GEP in a software package (Ferreira 2008) that we utilize here.

3. Case-study design

a. Motivation

Montane quantitative precipitation forecasts (QPFs) are difficult, yet are critically important for predicting flash floods, avalanches, landslides, and hydroelectric generation. For our short-range limited-area ensemble forecasts, we tried Kalman filter postprocessing and neural network postprocessing on the individual ensemble members, but both were unsuccessful. The Kalman filter often learns of a QPF bias on a rainy day, and inappropriately applies that bias correction on the next day even if no rain is forecast. The neural network creates a time

series with the correct climatological frequency of rain events, but the individual rain forecasts do not match the actual rain events. Also, the neural network does not always find the global minimum of its cost function. Moving averages were found to offer some bias correction skill for the precipitation degree-of-mass balance in the mountains of western Canada (McCollor and Stull 2008a). To explore alternatives, we turned to GEP, initially focused on only statistical postprocessing.

While experimenting with GEP, we realized that the optimum algorithm found by GEP was combining the raw NWP precipitation forecasts while simultaneously removing bias. Namely, it was producing a bias-corrected DEF in one step.

The resulting algorithms that GEP created were nothing like we had ever seen—having a complexity that we would never have created by hand. Yet they worked surprisingly well for some of the weather stations we tested. Also, they avoided the overfitting problem that plagues some neural network solutions. Based on these enticing results, we designed a more extensive test of this bias-corrected DEF method for many observation stations over many months using multimodel NWP ensemble output, as described below.

b. Numerical models

The NWP models used at UBC for daily limited-area, short-range (60 h) runs are

- the Weather Research and Forecasting model (WRF; Skamarock et al. 2005)
- the fifth-generation Pennsylvania State University–National Center for Atmospheric Research Mesoscale Model (MM5; Grell et al. 1994), and
- the Mesoscale Community Compressible model (MC2; Benoit et al. 1997).

These models are run in self-nested mode at the following horizontal grid spacings with the following grid designations: grid 1 (g1) is 108 km, grid 2 (g2) is 36 km, grid 3 (g3) is 12 km, grid 4 (g4) is 4 km (but no grid 4 for WRF). Thus, on days when all models ran successfully, our multimodel multiresolution ensemble had 11 members. WRF and MM5 run with two-way nesting, and MC2 with one-way nesting. For the MC2 model, each subsequent finer grid is started 3 h after the coarser grid, to give time for the numerical solution of the coarser grid to stabilize before starting the finer grid, and to allow the modeled precipitation to spin up.

Initial and boundary conditions for the 108-km runs came from the National Centers for Environmental Prediction’s (NCEP) North American Mesoscale (NAM) 0000 UTC runs. Sixty-hour forecasts are made, but only the 24-h period between model forecast hours 12 and 36 are

FIG. 4. Location of weather stations in southwestern BC, Canada.

used, corresponding to 1200 UTC on day 1 and 1200 UTC on day 2.

c. Observation data

The case study area covers southwestern British Columbia, Canada, along the Pacific coast of North America (Fig. 4). This area of complex terrain is characterized by high mountains (many peaks of 2000–3000 m), deep narrow valleys, coastlines, fjords, glaciers, and one large river delta near sea level where metropolitan Vancouver is located. One mountain range forms a spine of Vancouver Island. The higher Coast Range Mountains are parallel to Vancouver Island, just east of the Georgia Strait that separates Vancouver Island from the mainland. This region experiences frequent land-falling Pacific cyclones in winter, some with strong pre-frontal low-altitude jets that transport copious amounts of moisture from the tropics (called the pineapple express).

Twenty-four surface weather stations in this region are used (Fig. 4, Table 1). Twenty-one of the stations are operated by the BC Hydro hydroelectric corporation as private stations with no International Civil Aviation Organization (ICAO) identifier. Three of the stations [Canadian Forces Base Comox, near Comox, British Columbia (CYQQ); Vancouver International Airport (CYVR), and Abbotsford International Airport, in Ab-

botsford, British Columbia (CYXX)] are operated by Environment Canada and are abbreviated here as YQQ, YVR, and YXX, respectively. The dataset spans October 2003–March 2005, which includes two fall–spring rainy seasons. Summer data were excluded from this study. Any rainy season dates with missing observations or missing forecasts were removed, and the remaining forecast–observation pairs were quality controlled to remove unphysical values. For this case study, if any one or more ensemble members were missing on a date, we removed that date from the dataset.

The remaining dataset contained 198 days, of which 71% had nonzero precipitation, and (33%, 21%, 9%) had greater than (5, 10, 25) mm day^{−1}. The 198-day dataset was divided into three unequal sequential portions for training, testing, and scoring (Table 2). The reason for not interlacing these three subsets is that we wanted to test the best algorithm in simulated forecast mode; namely, for future days (i.e., the scoring set) that might have synoptic regimes that are different from those in the training and testing periods.

d. Procedure

The predictand is the difference bias (B24) of 24-h accumulated precipitation (P24) for the MC2 grid 4 forecast at any one weather station:

TABLE 1. Twenty-four surface weather stations in Vancouver Island and Lower Mainland with data for the period Oct 2003–Mar 2005. All stations are operated by BC Hydro except for the three Environment Canada stations set in boldface in the ID column.

ID	Name	Lat (°)	Lon (°)	Elevation (m)	Location
ALU	Alouette Dam Forebay	49.29	122.48	125	Lower mainland/interior
ASH	Alsie Lake	49.43	125.14	340	Vancouver Island
BCK	Bear Creek Reservoir	48.5	123.91	419	Vancouver Island
BLN	Brolorne Upper	50.8	122.75	1920	Lower mainland/interior
CLO	Clowham Falls	49.71	123.52	10	Lower mainland/interior
CMU	Upper Cheakamus	50.12	123.13	880	Lower mainland/interior
CMX	Comox Dam Forebay	49.64	125.09	135	Vancouver Island
CRU	Cruickshank River	49.57	125.2	150	Vancouver Island
ELA	Elaho River	50.22	123.58	206	Lower mainland/interior
ELK	ELK River above Campbell Lake	49.85	125.8	270	Vancouver Island
GLD	Gold River near Ucona River	49.71	126.11	10	Vancouver Island
GOC	Gold Greek	49.45	122.48	794	Lower mainland/interior
HEB	Heber River near Gold River	49.84	125.98	215	Vancouver Island
LJU	Downtown Upper	50.86	123.18	1829	Lower mainland/interior
MIS	Mission Ridge	50.75	122.24	1850	Lower mainland/interior
NTY	North Tyaughton Creek	51.13	122.76	1969	Lower mainland/interior
SCA	Strathcona Dam	49.98	125.58	249	Vancouver Island
STA	Stave River above Lake	49.55	122.32	330	Lower mainland/interior
STV	Upper Cheakamus	49.63	122.41	930	Lower mainland/interior
WAH	Jones Lake	49.23	121.62	641	Lower mainland/interior
WOL	Wolf River Upper	49.68	125.74	1490	Vancouver Island
YQQ	Comox	49.72	124.9	23	Lower mainland/interior
YVR	Vancouver International Airport	49.18	123.17	3	Lower mainland/interior
YXX	Abbotsford Airport	49.03	122.37	58	Lower mainland/interior

$$B24 = P24(\text{Observation}) - P24[\text{MC2}(\text{Grid } 4)]. \quad (1)$$

The predictors are the forecast 24-h accumulated precipitation amounts from all the models and grids (Fig. 5). Separate GEP runs are made for each weather station, yielding completely different DEF algorithms at each. The 24-h accumulated precipitation variable was chosen for two reasons: 1) it has strong economic value to hydroelectric managers and 2) it has greater predictability than shorter precipitation intervals (Wandishin et al. 2001; Stensrud and Yussouf 2007; McCollor and Stull 2008a–c).

For any one station, the GEP population is trained using all available data from the *first* rainy season (107 forecast–observation pairs). These data are called the training set. After a stable population is reached (namely, after fitness scores plateau during the evolution), all of the surviving individuals are tested against the first half of the remaining data (a testing set of 45–51 forecast–observation pairs, depending on the station). With such a testing set, we weed out those individuals that overfit the data or that are unnecessarily complex, allowing a “best” DEF algorithm to be selected. This one best algorithm is then further evaluated (scored) against the remaining independent data (40 forecast–observation pairs, called the scoring set; see Table 2). The resulting verification statistics for the scoring set given here are for the 24-h precipitation, not for the bias B24.

e. Gene-expression-programming specifications

The GEP software package (Ferreira 2008) is flexible regarding the size and nature of the chromosomes, the functions available, the creation of the numerical constants, the choice of the fitness statistic, the mutation and modification rates, etc. Although Ferreira gives some guidelines, the user must experiment via trial and error to find parameters that yield viable populations.

After much experimentation for YVR, we settled on the GEP parameters listed in the “final setting” column in Table 3. The final setting is not necessarily the absolute best one (which might take an infinite number of worlds/runs to discover), but it is one that gives an ensemble average that works well for many of the stations. A set of 80 mathematic and logic functions (see the

TABLE 2. Data subsets. The quality controlled dataset contains 198 days, which are divided into three unequal portions for training, testing, and scoring. These three sets help us to test the best algorithm in a simulated forecast mode, namely, for future months that might have synoptic regimes and global climate variations different from those in the training period.

	No. of events	Starting date	Ending date
Training set	107	11 Oct 2003	11 Oct 2004
Testing set	47–51	12 Oct 2004	31 Dec 2004
Scoring set	40	1 Jan 2005	11 Mar 2005

FIG. 5. Road map of the DEF procedure, done independently at each weather station. (a) First, we use GEP on a training set of data to *find* an algorithm for the bias B24 of 24-h accumulated precipitation. Predictors are raw 24-h precipitation forecasts (P24) from 11 ensemble members, where subscripts (C, M, W) represent the (MC2, MM5, and WRF) models, and subscripts 1–4 indicate grids 1–4. The predictand is the bias for only *one* of the ensemble members: MC2 grid 4 (corresponding to subscript C4). (b) Then, for the testing and scoring datasets, we use the algorithm to create a single deterministic forecast $P24_{FCST}$ from all the ensemble members. $P24_{OBS}$ is the observed 24-h precipitation.

supplemental information online) were used to build and modify the chromosomes.

Different choices in allowed parameters allow different DEF algorithms to be created. Even with the same set of GEP parameters, different populations evolve when the GEP run is restarted, because of the quasi-random nature of the initialization, mutation, and selection mechanisms. This nonunique and irreproducible nature of GEP is disconcerting at first, but is a characteristic that is irrele-

vant to the ability of GEP to find useful economically valuable DEFs that work easily and effectively in an operational NWP environment (see section 6).

To illustrate the effects of different GEP parameter settings (Table 3) on the DEF algorithm for 24-h precipitation, separate evolutions (worlds or runs) are created for YVR, but with a smaller set of functions (+, −, ×, /, sqrt, exp, ln, x^2 , x^3 , $x^{1/3}$, sin, cos, and atan) for only the comparison runs prescribed by Table 3. After many runs

TABLE 3. GEP parameter settings. RRSE and RAE are fitness functions (see the appendix). See the supplemental material online for descriptions of the various genetic modification rates. Differences among settings I–III are highlighted in boldface.

	Final setting	Setting I	Setting II	Setting III
Population size	40	30	30	30
No. of elements in the training sample	107	107	107	107
No. of elements in the testing sample	51	51	51	51
No. of elements in the scoring samples	41	41	41	41
No. of functions	80	13	13	13
Terminal sets	11	11	11	11
Head length	15	10	10	10
No. of genes	7	4	4	2
Linking function between genes	Addition	Addition	Addition	Addition
Mutation rate	0.044	0.044	0.044	0.044
Inversion rate	0.1	0.1	0.1	0.1
Insertion sequence transposition rate	0.1	0.1	0.1	0.1
Root insertion sequence transposition rate	0.1	0.1	0.1	0.1
Gene transposition rate	0.1	0.1	0.1	0.1
One-point recombination rate	0.3	0.3	0.3	0.3
Two-point recombination rate	0.3	0.3	0.3	0.3
Gene recombination rate	0.1	0.1	0.1	0.1
Fitness function	RRSE	RAE	RRSE	RRSE

for each setting, the best bias-corrected-DEF algorithms are selected using the training and testing subsets of the data and are then verified against the scoring subset of the data. Figure 6 compares the verification scores for the best algorithms from these different GEP settings. Shown

for comparison is the verification for our current operational method, where a “pooled ensemble” (PE) average is created by weighting all ensemble members equally.

Our “final setting” yields a DEF with the best verification scores for all statistics except mean error. It also

FIG. 6. Verification statistics for GEP deterministic ensemble forecasts at the Vancouver airport, for the “scoring” subset of data. Four different GEP parameter settings (see Table 3) are shown: the final setting (black), setting I (light gray downward diagonal stripes), setting II (dark gray upward diagonal stripes), and setting III (dark gray horizontal stripes). Shown for comparison are statistics for the equally weighted average pooled ensemble (gray). Plotted are the 24-h accumulated precipitation ME (mm, zero is best), MAE (mm, zero is best), and RMSE (mm, zero is best). The correlation coefficient between forecasts and observations (r), and the DMB have been multiplied by 5 to make them more visible on the plot; hence, values closer to 5 are best.

has better correlation and a better “degree of mass balance” (DMB). DMB is defined as the ratio of the predicted to observed net water mass during the study period (Grubisic et al. 2005)—an important measure for hydrometeorologic studies (see the appendix). Changing the fitness function from the RRSE to the RAE verifies significantly worse (cf. settings I and II in Fig. 6). Also, the size of genome (the allowed complexity of the algorithm) is very important. The effects of genome size can be examined by changing the number of genes or the lengths of the genes (cf. settings I and III).

Overall, the final setting yields DEFs that verify better than other settings we tried (not shown here). This does not mean it is the absolute best setting; however, it is better than the others we tested. These final settings are used for all case study and sensitivity experiments, which are described next.

4. Results

a. Verification for all 24 weather stations

In this case study, we performed GEP evolutions separately for each of the 24 weather stations, yielding different “best” DEF algorithms for each station (see the supplemental information online for the best algorithm for YVR). As described in the previous section, training and testing portions are used to find the best DEF algorithm, and the scoring portion is used for all verification statistics below.

Detailed verification statistics are shown in Fig. 7, and a concise tally of our results is in Fig. 8. The different statistical measures give different indications as to which DEF approach (GEP or PE) is best. Since root-mean-square error (RMSE) is very similar to the RRSE statistic that was used in the “training” set to determine fitness, it is anticipated that RMSE will show good GEP results at the largest number of stations. The RRSE fitness measure favors extreme precipitation events.

GEP gives lower (better) RMSEs than does the classical PE method for 11 of the 24 stations and is nearly tied for 10 stations, while the PE method is better for 3 stations. Using the MAE metric, only 6 of the 24 stations have lower errors using GEP, 5 stations are nearly tied, and 13 stations have lower errors using PE. Although some stations were not well fit by the GEP algorithms that were found using the “final” settings from the previous section, it is possible that better GEP DEF algorithms could be found with different settings and variables. There is no reason that a single set of GEP settings should work the best for all stations, so future work will examine optimizing the GEP settings separately for each station.

These results suggest that different DEF methods are best at different stations, with no clear winner for this case study dataset. Using an overall tally as a crude measure, Fig. 8 has 41 black tiles (GEP best), 41 white tiles (pooled-ensemble best), and 38 gray tiles (nearly equal verification). Comparing the stations in Fig. 8 with their locations in Fig. 1, one finds that the stations for which GEP DEFs are best are scattered over a wide range of terrains and distances from the coast. There is a cluster of good GEP stations in the Lower Fraser Valley [YVR, YXX, and Alouette Dam, near Maple Ridge, British Columbia (ALU)]—the metropolitan Vancouver region.

b. A closer examination of Vancouver airport station (YVR)

The Vancouver International Airport station gets special scrutiny because of the large number of people (2 million) in the metropolitan Vancouver area who can be impacted by urban flooding and landslides triggered by heavy-precipitation events. Figure 9 shows 24-h precipitation *amount* observed at YVR over a period spanning the two rainy seasons. Figure 10 (data points) shows the precipitation *bias* (the predictand) of the MC2 model at 4-km grid spacing for that same period. Also shown in Fig. 10 is the GEP estimate of the 24-h precipitation bias. One component of our implementation is that whenever the GEP bias-corrected forecast gives a negative precipitation amount, we truncate the amount to zero. Thus, those “apparent” erroneously large negative biases (B24) of the GEP forecasts (in Figs. 10a and 10b) *do*, in fact, give accurate precipitation amount (P24) forecasts after truncation.

A comparison of DEF precipitation amount (P24) forecasts is shown in Fig. 11a for the scoring data subset at the Vancouver airport. To get these results, GEP bias forecasts (Fig. 5a) are applied to the raw MC2(grid 4) forecasts using Eq. (1) to get forecasts of precipitation amounts (Fig. 5b). Figure 11b shows large variability among the raw NWP forecasts (ensemble members, used as predictors) for the heavy-rain event in Fig. 11a. For this rain event on 19 January 2005, the pooled ensemble better fits the observation than the GEP forecast, while during 20 January–5 February 2005 the GEP forecasts fit better (Figs. 11a and 11b). For future research, we will use GEP to first classify the predicted precipitation (extreme, normal, none) before finding the best DEF QPFs for each category.

5. Sensitivity studies to identify the most important ensemble members

The DEF algorithms devised by GEP to combine all the ensemble members are very complicated (see the

FIG. 7. Verification statistics of 24-h accumulated precipitation forecasts for the 24 stations in southwestern BC. Black is for GEP, and white for PE (pooled ensemble of equally weighted ensemble members): for (a) RMSE, (b) ME, (c) MAE, (d) correlation (r), and (e) DMB. For (a)–(c) an error of zero is best. For (d), (e) a value of one is best.

	RMSE	ME	MAE	Correlation	DMB		RMSE	ME	MAE	Correlation	DMB
ALU						HEB					
ASH						LJU					
BCK						MIS					
BLN						NTY					
CLO						SCA					
CMU						STA					
CMX						STV					
CRU						WAH					
ELA						WOL					
ELK						YQQ					
GLD						YVR					
GOC						YXX					

FIG. 8. Comparison between GEP and PE verification statistics at all 24 weather stations. Black indicates situations where GEP gives a better result, gray indicates no clear winner, and white represents situations where PE gives a better result.

supplemental information online). While they utilize all 11 ensemble members as predictors, it is not obvious from the equations which members have the largest influence on the DEF. One way to address this is via simple sensitivity studies.

Keeping the GEP algorithm fixed for each station, we input a large error for one of the predictors (i.e., one of the ensemble members), and measure the resulting

verification statistic for P24. If the ensemble member was relatively unimportant, then a large error in its P24 value will have only a small effect on the overall verification. However, if the ensemble member is an important contributor to the overall GEP forecast, then the introduction of a large error into the input ensemble member will result in a substantial increase in the verification statistic error for the whole GEP forecast.

FIG. 9. Time series of 24-h precipitation amount observations for YVR during the period 10 Jan 2003–15 Mar 2005. Gray lines at the bottom of the figure show the dates of all of available pairs of observations and forecasts. Dash lines divide the dataset into three subsets: (a) training, (b) testing, and (c) scoring. The unlabeled section between (a) and (b) is the drier summer season (not part of this study).

FIG. 10. GEP ensemble estimates (gray line) of the forecast bias B_{24} (data points) as defined by Eq. (1), for the (a) training, (b) testing, and (c) scoring subsets of data for YVR.

As an example of this sensitivity study, we halve and double the ensemble member inputs one at a time, and record the resulting verification statistics (Fig. 12) for the Vancouver airport. (Because the GEP algorithm can be highly nonlinear, doubling and halving could yield different conclusions. As it turns out, halving and doubling gave identical results.)

We find that a small number of “important” ensemble members dominate the DEF outcome. The set of important predictors has the following characteristics: 1) many of the models are included (e.g., MC2, MM5, WRF), 2) a range of grid sizes are included (but not necessarily from the same model), and 3) the set contains a small number of elements that satisfy characteristics 1 and 2. For example, Fig. 12 shows that the following predictors were important for a good GEP DEF algorithm at YVR: WRF (108 km), MM5 (36 km), MM5 (108 km), and MC2 (4 km).

A fourth characteristic is that the set of important predictors often changes from station to station, but continues to satisfy characteristics 1–3. This difference is

not surprising, given that GEP gives different algorithms for each different station. To illustrate this point, Tables 4a and 4b show the “ensemble space” of our case study, and compare the important ensemble members for YVR and YQQ.

Thus, the DEF that is most efficient (providing the maximum information per fewest ensemble members) is gained by spanning all grid sizes (which helps compensate for the location errors of synoptic systems, and for terrain-related errors) and spanning all models (maximizing the variety of model physics, dynamics, and discretizations). Any additional ensemble members for models or grids already covered by the important set offer marginal new information. Namely, not all of the ensemble members in a multimodel multi-grid-size DEF provide independent data. Thus, an efficient DEF spans the ensemble space sparsely. Similar conclusions have been found using best-member diagrams (Hamill and Colucci 1997; Charles and Colle 2009) and Bayesian model-averaging weights (Raftery et al. 2005).

FIG. 11. (a) Forecasts of the 24-h accumulated precipitation amount for YVR from the NWP pooled ensemble (dark gray dashes with crosses) and GEP (light gray line and diamonds). Dots show the observed precipitation amount, P24. (b) Illustration of forecasts from each of the ensemble members for 3 days during the peak rain event. Also plotted in (b) are the observations and two different DEF formulations (PE and GEP).

To examine the robustness of this result, we ran additional GEP evolutions where the ensemble members were reduced from all 11 to the 3 most important ones indicated in Table 4a. The DEF verification scores (Fig. 13) at YVR using the three “best” ensemble members are only slightly degraded from the scores using all 11 members.

We repeated experiment with a variety of different three best ensemble members (based on other individuals from the GEP run that were almost as good as the best), and also based on the pooled ensemble (with 3 and

with 11 members; see Fig. 13). For this case study at YVR, all GEP DEFs (with 3- or 11-member ensembles) give better forecasts than all pooled ensembles (with 3 or 11 members). These results support the hypothesis that sparse spanning of ensemble space can capture most of the signal, if the ensemble members are chosen wisely.

In future work, we will examine the weaker ensemble members to determine what aspects of the NWP dynamics or physics were inadequate, with the aim of suggesting improvements to the NWP codes.

FIG. 12. Sensitivity study for YVR, based on (a) halving the input signals to each GEP ensemble member separately, and (b) doubling the inputs. Error statistics, scaling, and interpretation are as in Fig. 6. All ensemble members show nearly the same skill as the original GEP except for WRF-G2, MM5-G2, MM5-G1, and MC2-G4, which have significant increases in their errors. These “important” ensemble members dominate the outcome of the whole DEF.

6. Discussion

a. Deterministic ensemble forecasts versus ensemble averages

When GEP is used to combine the ensemble members into one best deterministic ensemble forecast (DEF), the resulting algorithm is mathematically nothing like an “average” (see the supplemental information online). Hence, the ensemble average is just one of many ways to combine ensemble members into a deterministic forecast.

Probability and ensemble spread information is lost when the GEP creates the single deterministic forecast. However, there are other ways to use GEP to preserve ensemble spread information. For example, it may be used for statistical postprocessing each ensemble mem-

ber separately. Then, the resulting GEP-corrected ensemble members can be combined into probabilistic forecasts using traditional ensemble-averaging methods or using nonlinear combinations.

b. Reproducibility, uniqueness, and complexity

GEP creates irreproducible algorithms. Namely, different GEP runs for the same station and same training set, and using the same GEP parameters, will evolve toward different DEF algorithms, the best of which verify nearly equally well. This is a by-product of intentionally introducing random numbers into the specification of the initial population and into the mutation and survival of competing algorithms as they evolve. For modern science founded on experiments that can be reproduced in different laboratories, the irreproducibility

TABLE 4. A map of the “ensemble space” for our case study, showing the 11 members of this multimodel, multigrid ensemble. Those ensemble members that are most important to the deterministic ensemble forecast appear bold and underlined. Less important members are indicated in a italic font. (a) The results for YVR. The MM5(grid 1) was moderately important and appears set in boldface. (b) The results for YQQ.

(a) YVR				
Model	Horizontal grid size (km)			
	108 (grid 1)	36 (grid 2)	12 (grid 3)	4 (grid 4)
MC2	<i>1</i>	<i>2</i>	<i>3</i>	<u>4</u>
MM5	5	6	<i>7</i>	<u>8</u>
WRF	9	<i>10</i>	<i>11</i>	<i>N/A</i>

(b) YQQ				
Model	Horizontal grid size (km)			
	108 (grid 1)	36 (grid 2)	12 (grid 3)	4 (grid 4)
MC2	<i>1</i>	<i>2</i>	<i>3</i>	<u>4</u>
MM5	5	<i>6</i>	<i>7</i>	<u>8</u>
WRF	9	<i>10</i>	<i>11</i>	<i>N/A</i>

of GEP algorithms can be disconcerting. But this irreproducibility is an intrinsic nature of evolution (Darwin 1859).

A reassuring outcome is that the verification scores of the best individual DEFs from different runs are often nearly identical. Namely, each different DEF algorithm is able to extract from the predictor whatever information is available that can be mapped onto the signal of the predictand. Hence, each successful, evolved DEF algorithm has nearly equal utility; that is, they can improve the weather forecasts. In this sense, the successful algorithms are reproducible in their ability to produce the same outputs (within some error tolerance) for identical inputs, even though their internal makeups are different and irreproducible.

GEP produces nonunique algorithms. This is illustrated in Fig. 3, where curves a and b had different sets of functions available during their evolutions, and evolved toward radically different algorithms. Yet each fit the noisy data equally well. This robustness in the face of non-uniqueness highlights the power of evolutionary computing, and again points to the utility of the end result.

GEP produces DEF algorithms that are much more complex than most humans would have created. Part of this complexity is artificial and redundant, such as is illustrated by the GEP equation in Fig. 2b. In that example, the equation contains factors of the form $x/(x + x)$, which is how that GEP run created a needed constant value of 0.5. Different runs might have created that constant with different formulas. Although we manually simplified that equation (Fig. 2b) into a form pleasing to the human eye, the GEP equation is just as valid, and both yield the same outputs for the same inputs (except that more roundoff errors can accumulate when un-

FIG. 13. Verification statistics for the GEP bias-corrected ensembles at YVR for the six different ensemble runs. From left to right (light shading to dark), 1) GEP with three members (MC2 – G3, MM5 – G2, and WRF – G1), 2) GEP with three members (MC2 – G4, MM5 – G2, and WRF – G1), 3) the original GEP with all 11 members, 4) the pooled ensemble with three members (MC2 – G3, MM5 – G2, and WRF – G1), 5) the pooled ensemble with three members (MC2 – G4, MM5 – G2, and WRF – G), and 6) the pooled ensemble with all 11 members. Error statistics, scaling, and interpretation are as in Fig. 6.

necessary computations are performed). Even when parsimony pressure is used to reward the simpler algorithms with greater fitness scores, the procedure to reach the final “simplified” algorithm is nonetheless evolutionary, and only sometimes reaches the simplified equation that a human might have created. The cost of this redundant complexity is increased computation time every day when it is used to compute the DEF.

But another part of the complexity is valuable, because it might fit the desired signal better than a simpler algorithm. In spite of this value, the complexity can be daunting, as shown for YVR in the supplemental information online.

The decision of when to end a GEP run is arbitrary—another disconcerting characteristic. In some simple situations, such as for the noisy but simple data in Figs. 2 and 3, the evolution often reaches the maximum fitness possible without overfitting (i.e., without trying to fit the noise). In this case, it is obvious when to stop the evolution. But for more complex and realistic meteorological problems, the point at which to end the evolution is not obvious. When the evolution starts, significant improvements to the “best” individual algorithm are frequent and obvious. But as evolution proceeds, improvements happen less frequently and often contribute less increase in fitness. After 1000 or 10 000 additional generations with no change, one might be inclined to end the GEP run, even though there is a chance that the 10 001st generation might have offered significant improvement. If the resulting

DEF algorithm is good enough (i.e., has small enough error in the independent scoring dataset), then we are done. If the error is still large after stopping the evolution, then a recommended approach is to start a new GEP run from scratch, namely, create a new world that will evolve the population of algorithms to a different ending state. Ferreira (2006) thoroughly discusses the probability of reaching a useful ending state via multiple GEP runs.

c. Operational NWP usage of complex GEP algorithms

Is this postprocessing method worth the added complexity? This is a subjective question for which different users will reach different conclusions. For our own daily operational NWP at UBC, it is worth the added complexity, but only at those weather stations where GEP gives better results. After spending hours of computations daily on hundreds of processors to produce raw NWP output, it is trivial to spend seconds of computation on a couple processors to solve the best DEF algorithm (previously evolved by GEP) to improve forecasts at hundreds of weather station locations in British Columbia. The amount of forecast improvement directly translates into an improved ability to predict potential hydroelectric generation, flash floods, snow avalanches, and water-driven landslides in the mountains.

What about the time spent to create the evolved algorithm in the first place? The algorithms are created once, but used many times. Even if they need to be reevolved every season for every city, the utility is large compared to the expense and time needed to create them. Also, the GEP runs can be automated. Again, it is a subjective decision; at UBC, the value is worth the expense.

How will GEP perform for other types of ensembles? We tested only multimodel ensembles here, which might be capturing more of the initial uncertainty in the forecast. For single-model ensembles with multiple initial conditions, the performance of GEP is untested.

7. Conclusions and future work

For our case study data, we find that GEP DEFs are better than equally weighted ensemble averages of precipitation for about half of the mountain weather stations tested. A sparse sampling of the multimodel multigrid-size “ensemble space” spanned by the ensemble members can create a DEF almost as good as the full 11-member ensemble. The best DEF algorithms found using GEP are nonunique and irreproducible, yet can give consistent results that can be used to good ad-

vantage at operational forecast centers. In spite of the relative complexity of the DEF algorithms created using GEP, they are noniterative and, thus, computationally efficient to use.

For some weather stations, the DEF algorithms found by GEP do not give better results than those from the pooled (equally weighted) ensemble average. There is no law of computing that says you must use the same bias-corrected ensemble methods at all weather station locations, so we recommend the use of different methods at different stations as appropriate.

We plan the following future research. We will test GEP DEFs over additional datasets, other locations, and for other weather elements. We will include as predictors the raw NWP precipitation forecasts from neighboring grid points and at neighboring times, to compensate for systematic timing or location biases in cyclone landfalls. We will also test using NWP forecasts of other weather elements (temperature, humidity, winds, etc.) as predictors in the DEF algorithm for precipitation, and we will explore the use of different GEP parameter settings at different stations. We will also build on the work of Cannon (2008) and others to explore whether GEP can be used in two stages: first to classify the forecast precipitation (extreme, normal, or none), and then to use different DEF algorithms for the extreme versus normal events.

In addition to finding DEFs, we will apply GEP to each ensemble member individually strictly as a bias-correction method, yielding a set of ensemble members that can be used for probabilistic forecasts. We will compare GEP DEFs with additional regression and ensemble-averaging approaches, such as Bayesian model averaging. We will examine the weaker ensemble members to determine what aspects of the NWP dynamics or physics were inadequate, with the aim of suggesting improvements to the NWP codes. All of this work has the ultimate goal of improving the skill of weather forecasts over complex terrain.

Acknowledgments. Doug McCollor of BC Hydro provided the quality controlled dataset, and additional data came from Environment Canada. Thomas Nipen, George Hicks, and anonymous referees suggested important improvements. Funding was provided by the Canadian Natural Science and Engineering Research Council, the Canadian Foundation for Climate and Atmospheric Science, BC Hydro, the Canadian Foundation for Innovation, the British Columbia Knowledge Development Fund, and UBC. The Geophysical Disaster Computational Fluid Dynamics Center provided computing resources.

APPENDIX

Statistical Measures

The following statistical functions are used to determine the fitness (f) of individual algorithms (Ferreira 2006). For data index i between 1 and n data points, let E_i be the value estimated by the evolved algorithm, and O_i be the target value. For some of our experiments, $(E, O) = (\text{DEF bias-corrected precipitation amount, observed precipitation amount})$, while for other experiments, $(E, O) = (\text{DEF precipitation bias, precipitation bias between observations and the MC2 4-km forecast})$.

We find the following for the root relative squared error:

$$\text{RRSE} = \left[\frac{\sum_{i=1}^n (E_i - O_i)^2}{\sum_{i=1}^n (\bar{O} - O_i)^2} \right]^{1/2}, \quad (\text{A1})$$

where $\bar{O}$ is the mean target value, averaged over all i . The error in the DEF estimate is “relative” to the error that would have occurred if the estimate was just the average of the observations (namely, the climatology of the training set). From this error, the fitness is

$$f = 1000(1 + \text{RRSE})^{-1}, \quad (\text{A2})$$

where a perfectly fit individual has $f = 1000$ and f approaches zero as RRSE increases toward infinity. When parsimony pressure is included, (A2) is modified by adding a very small term that increases as the size of the chromosome decreases (see Ferreira 2006 for details).

The mean absolute error is

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |E_i - O_i|, \quad (\text{A3})$$

with fitness

$$f = 1000(1 + \text{MAE})^{-1}. \quad (\text{A4})$$

The relative absolute error is

$$\text{RAE} = \frac{\sum_{i=1}^n |E_i - O_i|}{\sum_{i=1}^n |\bar{O} - O_i|}, \quad (\text{A5})$$

with fitness

$$f = 1000(1 + \text{RAE})^{-1}. \quad (\text{A6})$$

For verification, we confirm the estimated precipitation against the paired 24-h accumulated precipitation observation. Even for those algorithms evolved for forecasting precipitation bias, we first convert the estimated bias to a precipitation amount using Eq. (1) before verifying against the observed precipitation. In addition to MAE, the following statistical measures (Wilks 2006) are used for verification.

The degree of mass balance (DMB) is the ratio of predicted to observed net water mass over the whole verification period (Grubisic et al. 2005):

$$\text{DMB} = \frac{\sum_{i=1}^n E_i}{\sum_{i=1}^n O_i}. \quad (\text{A7})$$

For the perfect mass balance, $\text{DMB} = 1$.

The mean error is

$$\text{ME} = \frac{1}{n} \sum_{i=1}^n E_i - \frac{1}{n} \sum_{i=1}^n O_i = \bar{E} - \bar{O}, \quad (\text{A8})$$

where zero mean error is best. The overbar denotes the average over all data indices.

The root-mean-square error is

$$\text{RMSE} = \left[\frac{1}{n} \sum_{i=1}^n (E_i - O_i)^2 \right]^{1/2}, \quad (\text{A9})$$

where zero error is best. This error measure gives more weight to the outliers than does MAE.

The Pearson correlation coefficient is

$$r = \frac{\sum_{i=1}^n [(E_i - \bar{E})(O_i - \bar{O})]}{\left[\sum_{i=1}^n (E_i - \bar{E})^2 \sum_{i=1}^n (O_i - \bar{O})^2 \right]^{1/2}}, \quad (\text{A10})$$

where $r = 1$ indicates that the estimate and the observation vary together perfectly.

REFERENCES

- Adami, C., 1998: *Introduction to Artificial Life*. Springer/Telos, 374 pp.
- Benoit, R., M. Desgagne, P. Pellerin, S. Pellerin, and Y. Chartier, 1997: The Canadian MC2: A semi-Lagrangian, semi-implicit

- wideband atmospheric model suited for finescale process studies and simulation. *Mon. Wea. Rev.*, **125**, 2383–2415.
- Bozic, S. M., 1994: *Digital and Kalman Filtering: An Introduction to Discrete-Time Filtering and Optimum Linear Estimation*. 2 ed. Edward Arnold, 160 pp.
- Cannon, A. J., 2008: Probabilistic multisite precipitation downscaling by an expanded Bernoulli-gamma density network. *J. Hydrometeorol.*, **9**, 1284–1300.
- Charles, M. E., and B. A. Colle, 2009: Verification of extratropical cyclones within the NCEP operational models. Part II: The Short-Range Ensemble Forecast system. *Wea. Forecasting*, **24**, 1191–1214.
- Cheng, W. Y. Y., and W. J. Steenburgh, 2007: Strengths and weaknesses of MOS, running-mean bias removal, and Kalman filter techniques for improving model forecasts over the western United States. *Wea. Forecasting*, **22**, 1304–1318.
- Cramer, N. L., 1985: A representation for the adaptive generation of simple sequential programs. *Proceedings of the First International Conf. on Genetic Algorithms and Their Applications*, J. J. Grefenstette, Ed., Erlbaum, 183–187.
- Darwin, C., 1859: *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. John Murray, 502 pp.
- Delle Monache, L., T. Nipen, X. Deng, Y. Zhou, and R. Stull, 2006: Ozone ensemble forecasts: 2. A Kalman filter predictor bias correction. *J. Geophys. Res.*, **111**, D05308, doi:10.1029/2005JD006311.
- Eckel, F. A., and C. F. Mass, 2005: Aspects of effective mesoscale, short-range ensemble forecasting. *Wea. Forecasting*, **20**, 328–350.
- ECMWF, cited 2008: WG 3: Verification and applications of ensemble forecasts. Workshop on Ensemble Prediction Working Group 3 Rep., ECMWF, 6 pp. [Available online at http://www.ecmwf.int/newsevents/meetings/workshops/2007/ensemble_prediction/wg3.pdf.]
- Ferreira, C., 2001: Gene expression programming: A new adaptive algorithm for solving problems. *Complex Systems*, **13**, 87–129.
- , 2006: *Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence*. 2nd ed. Springer, 478 pp.
- , cited 2008: GeneXproTools_{4.0}: Modeling made easy. [Available online at <http://www.gepsoft.com>.]
- Gel, Y. R., 2007: Comparative analysis of the local observation-based (LOB) method and the nonparametric regression-based method for gridded bias correction in mesoscale weather forecasting. *Wea. Forecasting*, **22**, 1243–1256.
- Goldberg, D. E., 1989: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 412 pp.
- Grell, G., J. Dudhia, and D. R. Stauffer, 1994: A description of the fifth-generation Penn State/NCAR Mesoscale Model (MM5). NCAR Tech. Rep. TN-398+STR, 121 pp.
- Grubisic, V., R. K. Vellore, and A. W. Huggins, 2005: Quantitative precipitation forecasting of wintertime storms in the Sierra Nevada: Sensitivity to the microphysical parameterization and horizontal resolution. *Mon. Wea. Rev.*, **133**, 2834–2859.
- Hacker, J., and D. L. Rife, 2007: A practical approach to sequential estimation of systematic error on near-surface mesoscale grids. *Wea. Forecasting*, **22**, 1257–1273.
- Hamill, T. M., and S. J. Colucci, 1997: Verification of Eta-RSM short-range ensemble forecasts. *Mon. Wea. Rev.*, **125**, 1312–1327.
- , S. L. Mullen, C. Snyder, Z. Toth, and D. P. Baumhefner, 2000: Ensemble forecasting in the short to medium range: Report from a workshop. *Bull. Amer. Meteor. Soc.*, **81**, 2653–2664.
- Hansen, B., 2007: A fuzzy-logic based analog forecasting system for ceiling and visibility. *Wea. Forecasting*, **22**, 1319–1330.
- Holland, J. H., 1975: *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 183 pp.
- Jones, M. S., B. A. Colle, and J. S. Tongue, 2007: Evaluation of a mesoscale short-range ensemble forecast system over the northeast United States. *Wea. Forecasting*, **22**, 36–55.
- Kalnay, E., 2003: *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge University Press, 341 pp.
- Koza, J. R., 1992: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. The MIT Press, 840 pp.
- , 1994: *Genetic Programming II: Automatic Discovery of Reusable Programs*. The MIT Press, 768 pp.
- , F. H. Bennett III, D. Andre, and M. A. Keane, 1999: *Genetic Programming III: Darwinian Invention and Problem Solving*. Morgan Kaufmann, 1154 pp.
- , M. A. Keane, M. J. Streeter, W. Mydlowec, J. Yu, and G. Lanza, 2003: *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*. Springer, 590 pp.
- Lenski, R. E., C. Ofria, R. T. Pennock, and C. Adami, 2003: The evolutionary origin of complex features. *Nature*, **423**, 139–144.
- McCollor, D., and R. Stull, 2008a: Hydrometeorological accuracy enhancement via postprocessing of numerical weather forecasts in complex terrain. *Wea. Forecasting*, **23**, 131–144.
- , and —, 2008b: Hydrometeorological short-range ensemble forecasts in complex terrain. Part I: Meteorological evaluation. *Wea. Forecasting*, **23**, 533–556.
- , and —, 2008c: Hydrometeorological short-range ensemble forecasts in complex terrain. Part II: Economic evaluation. *Wea. Forecasting*, **23**, 557–574.
- , and —, 2009: Evaluation of probabilistic medium-range temperature forecasts from the North American Ensemble Forecast System. *Wea. Forecasting*, **24**, 3–17.
- Mitchell, T. M., 1997: *Machine Learning*. McGraw-Hill, 414 pp.
- Raftery, A. E., T. Gneiting, F. Balabdaoui, and M. Polakowski, 2005: Using Bayesian model averaging to calibrate forecast ensembles. *Mon. Wea. Rev.*, **133**, 1155–1174.
- Richardson, D. S., 2000: Skill and relative economic value of the ECMWF Ensemble Prediction System. *Quart. J. Roy. Meteor. Soc.*, **126**, 649–667.
- Skamarock, W. C., J. B. Klemp, J. Dudhia, D. O. Gill, D. M. Barker, W. Wang, and J. G. Powers, 2005: A description of the Advanced Research WRF version 2. NCAR Tech. Rep. TN-468+STF, 88 pp.
- Stensrud, D. J., and J. A. Skindlov, 1996: Gridpoint predictions of high temperature from a mesoscale model. *Wea. Forecasting*, **11**, 103–110.
- , and N. Yussouf, 2003: Short-range ensemble predictions of 2-m temperature and dewpoint temperature over New England. *Mon. Wea. Rev.*, **131**, 2510–2524.
- , and —, 2005: Bias-corrected short-range ensemble forecasts of near-surface variables. *Meteor. Appl.*, **12**, 217–230.
- , and —, 2007: Reliable probabilistic quantitative precipitation forecasts from a short-range ensemble forecasting system. *Wea. Forecasting*, **22**, 3–17.
- Wandishin, M. S., S. L. Mullen, D. J. Stensrud, and H. E. Brooks, 2001: Evaluation of a short-range multimodel ensemble system. *Mon. Wea. Rev.*, **129**, 729–747.
- Wilks, D. S., 2006: *Statistical Methods in the Atmospheric Sciences*. 2nd ed. Academic Press, 627 pp.

- Woodcock, F., and C. Engel, 2005: Operational consensus forecasts. *Wea. Forecasting*, **20**, 101–111.
- Yuan, H., X. Gao, S. L. Mullen, S. Sorooshian, J. Du, and H.-M. H. Juang, 2007: Calibration of probabilistic quantitative precipitation forecasts with an artificial neural network. *Wea. Forecasting*, **22**, 1287–1303.
- Yussouf, N., and D. J. Stensrud, 2006: Prediction of near-surface variables at independent locations from a bias-corrected ensemble forecasting system. *Mon. Wea. Rev.*, **134**, 3415–3424.
- , and —, 2007: Bias-corrected short-range ensemble forecasts of near-surface variables during the 2005/06 cool season. *Wea. Forecasting*, **22**, 1274–1286.
- Zhu, Y., Z. Toth, R. Wobus, D. Richardson, and K. Mylne, 2002: The economic value of ensemble-based weather forecasts. *Bull. Amer. Meteor. Soc.*, **83**, 73–83.