Intelligent Model Selection for Hillclimbing Search in Computer-Aided Design*

Thomas Ellman
John Keane
Mark Schwabacher
Department of Computer Science
Hill Center for Mathematical Sciences
Rutgers University
New Brunswick, NJ 08903
{ellman,keane,schwabac}@cs.rutgers.edu
CAP-TR-11

Abstract

Models of physical systems can differ according to computational cost, accuracy and precision, among other things. Depending on the problem solving task at hand, different models will be appropriate. Several investigators have recently developed methods of automatically selecting among multiple models of physical systems. Our research is novel in that we are developing model selection techniques specifically suited to computer-aided design. Our approach is based on the idea that artifact performance models for computer-aided design should be chosen *in light of the design decisions they are required to support.* We have developed a technique called "Gradient Magnitude Model Selection" (GMMS), which embodies this principle. GMMS operates in the context of a hillclimbing search process. It selects the simplest model that meets the needs of the hillclimbing algorithm in which it operates. We are using the domain of sailing yacht design as a testbed for this research. We have implemented GMMS and used it in hillclimbing search to decide between a computationally expensive potential-flow program and an algebraic approximation to analyze the performance of sailing yachts. Experimental tests show that GMMS makes the design process faster than it would be if the most expensive model were used for all design evaluations. GMMS achieves this performance improvement with little or no sacrifice in the quality of the resulting design.

*Also appears in the Proceedings of the Eleventh National Conference on Artificial Intelligence, Washington, D.C., July, 1993

1 Introduction

Models of a given physical system can differ along several dimensions, including the cost of using the model, the accuracy and precision of the results, the scope of applicability of the model and the data required to execute the model, among others. More than one model is often needed because different tasks require different tradeoffs among these dimensions. A variety of criteria and techniques have been proposed for selecting among various alternative models of physical systems. For example, some techniques select appropriate models by analyzing the structure of the query the model is intended to answer [Falkenhainer and Forbus, 1991], [Ling and Steinberg, 1992], [Weld and Addanki, 1991]. Another approach selects an appropriate model by reasoning about the simplifying assumptions underlying the available models [Addanki *et al.*, 1991]. Yet another approach reasons about the accuracy of the results the model must produce [Weld, 1991], [Falkenhainer, 1992].

We are are developing model selection techniques specifically suited to computer-aided design. Our approach is based on the idea that artifact performance models for computer-aided design should be chosen *in light of the design decisions they are required to support.* We have developed a technique called "Gradient Magnitude Model Selection" (GMMS), which embodies this principle. GMMS operates in the context of a hillclimbing search process. It selects the computationally cheapest model that meets the needs of the hillclimbing algorithm in which it operates.

Intelligent model selection is crucial for the overall performance of computer-aided design systems. The selected models must be accurate enough to ensure that the final artifact design is optimal with respect to some performance criterion, or else satisfactory with respect to specific performance objectives. The selected models must also be as computationally inexpensive as possible. Cheaper models enable a design system to spend less time on evaluation and more time on search. Broader search typically leads in turn to superior designs. These facts will remain true, even with the widespread use of supercomputers. The combinatorics of most realistic design problems are such that exhaustive search will probably never be feasible. There will always be an advantage in using the cheapest model that supports the necessary design decisions.

Model selection is a task that arises often in the day to day work of human design engineers. A human engineer's expertise consists, in part, of the ability to intelligently choose among various exact or approximate models of a physical system. In particular, as an engineer accumulates experience over his career, he learns which models are best suited to each modeling task he typically encounters in his work. This knowledge is one of the things that makes him an expert. Therefore, to the extent that GMMS successfully solves the model selection task, it automates a component of the computer-aided design process that is currently handled by human experts. GMMS may also be seen as a technique for attacking a standard AI problem: using knowledge to guide search. In particular, GMMS uses knowledge in the form of exact and approximate models, to guide hillclimbing design optimization. Related knowledge-based techniques for controlling numerical design optimization are described in [Cerbone and Dietterich, 1991] and [Tcheng *et al.*, 1991].

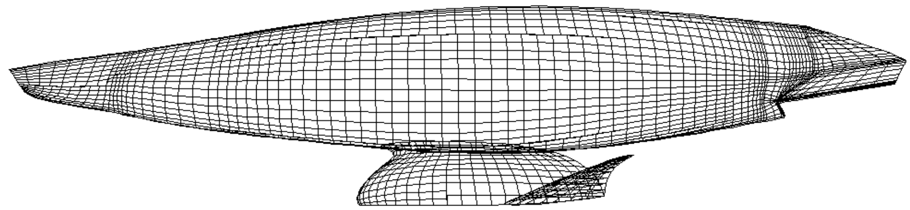

Figure 1: The Stars and Stripes '87 Hull

2 Yacht Design: A Testbed Domain

The GMMS technique has been developed and tested in the domain of 12-meter racing yachts, the class of yachts that race in the America's Cup competition. An example of a 12-meter yacht, the Stars and Stripes '87, is shown in Figure 1. This yacht won the America's cup back from Australia in 1987 [Letcher *et al.*, 1987]. Racing yachts can be designed to meet a variety of objectives. Possible yacht design goals include: *Course Time Goals*, *Rating Goals* and *Cost Goals*. In our research we have chosen to focus on a course time goal, i.e., minimizing the time it takes for a yacht to traverse a given race course under given wind conditions. Our system evaluates $CourseTime$ using a "Velocity Prediction Program", called "VPP". The organization of VPP is described in Figure 2. VPP takes as input a set of B-Spline surfaces representing the geometry of the yacht hull. Each surface is itself represented as a matrix of "control points" that define its shape. VPP begins by using the "hull processing models" to determine physically meaningful quantities impacting on the performance of the yacht, e.g., wave resistance (R_w), friction resistance (R_f), effective draft (T_{eff}), vertical center of gravity (Vcg) and vertical center of pressure (Zcp), among others. These quantities are then used in the "velocity prediction model" to set up non-linear equations describing the balance of forces and torques on the yacht. The velocity prediction model uses an iterative method to solve these equations and thereby determine the "velocity polar", i.e., a table giving the velocity of the yacht under various wind speeds and directions of heading. Finally, the "race model" uses the velocity polar to determine the total time to traverse the given course, assuming the given wind speed.

3 Hillclimbing for Design Optimization

Hillclimbing search is useful for attacking design optimization when the number of parameters is so large that exhaustive search methods are not practical. Our system uses steepest-descent as our basic hillclimbing method [Press *et al.*, 1986]. The steepest-descent algorithm operates by repeatedly computing the gradient of the evaluation function. (In the yacht domain, this requires computing the partial derivatives of $CourseTime$ with respect to each operator parameter.) The algorithm then takes a step in the direction of the gradient, and evaluates

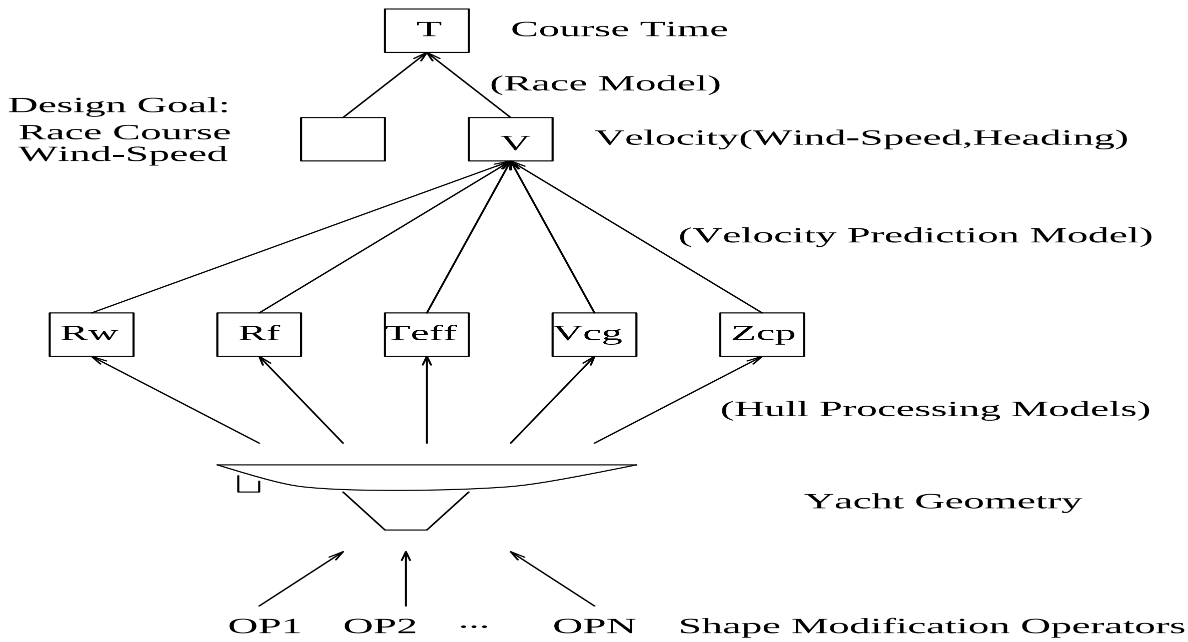

Figure 2: Velocity Prediction Program

the resulting point. If the new point is better than the old one, the new point becomes the current one, and the algorithm iterates. The algorithm terminates if the gradient is zero, or if a step in the direction of the gradient fails to improve the design.

A number of enhancements to the hillclimbing algorithm have been adopted to deal with practical difficulties arising in the yacht design domain. The program we use to compute $CourseTime$ (VPP) is a commercial software product. Nevertheless, it suffers from a number of deficiencies that make hillclimbing difficult. For example, it may return a spurious root of the balance of force equations that it solves. It may also exhibit discontinuities, due to numerical round-off error, or due to discretization of the (theoretically) continuous yacht hull surface. These deficiencies can produce "noise" in the evaluation function surface over which the hillclimbing algorithm is moving. The algorithm can easily get stuck at a point that appears to be a local optimum, but is nevertheless not locally optimal in terms of the true physics of the yacht design space. To overcome these difficulties, we have endowed the hillclimbing algorithm with some special features. To begin with, we arrange for the algorithm to use a range of different step sizes. The algorithm does not terminate until all of the step sizes fail to improve the design. The algorithm can therefore jump over hills of width less than the maximum step size. In addition, we provide the algorithm with an estimate of the magnitude of the noise in the evaluation function. The algorithm attempts to climb over any hills with height equal to the noise magnitude or lower. The resulting algorithm is more robust than the original algorithm.

4 Modeling Choices in Yacht Design

A number of modeling choices arise in the context of sailing yacht design. These choices are outlined in Figure 3. Probably the most important is the choice of models for estimating

- Algebraic Approximations v. Computational Fluid-Dynamics: The effective draft T_{eff} of a yacht can be estimated using an algebraic approximation or by using a potential flow code called "*PMARC*".
- Reuse of Prior Results v. Recomputation of Results: Some physical quantities may not change significantly when a design is modified. For a given physical quantity, its value may be retrieved from a prior candidate design, or its value may be recomputed from scratch.
- Linear Approximations v. Non-Linear Models: Velocity polars can be computed as linear functions of resistances and geometric quantities or by directly solving non-linear force and torque balancing equations.

Figure 3: Modeling Choices in Yacht Design

the effective draft (T_{eff}) of a yacht. Effective draft is a measure of the amount of drag produced by the keel as a result of the lift it generates. An accurate estimate of this quantity is quite important for analyzing the performance of a sailing yacht. Unfortunately, the most accurate way to estimate effective draft is to run a highly expensive potential flow code called *PMARC*. (This code takes approximately one hour when running on a Sun Microsystems Sparcstation 2 Workstation.) Effective draft can also be estimated using an algebraic approximation with the general form outlined below:

$$T_{eff} = K\sqrt{D^2 - 2A_{ms}/\pi}$$

$$D = Maximum \;\; Keel \;\; Draft$$

$$A_{ms} = Midship \;\; Hull \;\; Cross \;\; Section \;\; Area$$

This algebraic model is based on an approximation that treats a sailing yacht hull as an infinitely long cylinder and treats the keel of the yacht as an infinitely thin fin protruding from the cylinder. The constant K is chosen to fit the algebraic model to data obtained from wave tank tests, or from sample runs using the *PMARC* potential flow code. Although the algebraic approximation is comparatively easy to use, its results are not as accurate as those produced by the *PMARC* potential flow code.

Another important modeling choice involves the decision of when to reuse the results of a prior computation. The importance of this type of decision is illustrated by Figure 4. Suppose one is systematically exploring combinations of canoe-bodies and keels of a sailing yacht. In order to evaluate the performance of a yacht, one must evaluate the yacht's wave resistance R_w as well as its effective draft T_{eff}. Wave resistance depends mainly on the canoe-body of the yacht and is not significantly influenced by the keel. When only the keel is modified, wave resistance will not significantly change. Instead of recomputing wave resistance for the new yacht, the system can reuse the prior value. On the other hand, effective draft depends mainly on the keel of the yacht and is not significantly influenced by the canoe-body. When only the canoe-body is modified, effective draft will not significantly change. Instead of recomputing effective draft for the new yacht, the system can reuse the prior value. In fact, the entire matrix of yachts can be evaluated by computing wave resistance for a single row,

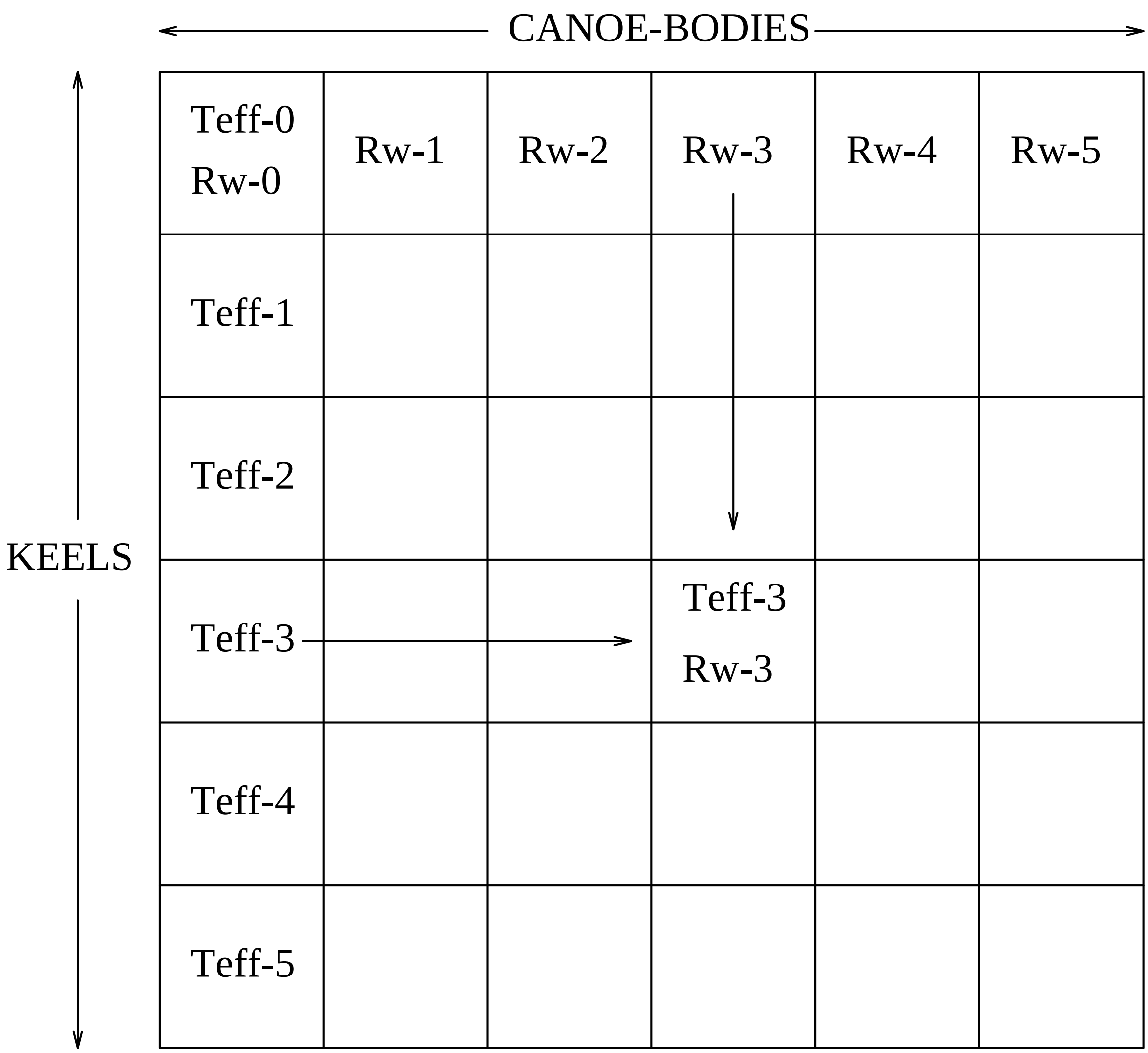

Figure 4: Reuse of Prior Results

and computing effective draft for a single column. By intelligently deciding when to reuse prior evaluation results, one can significantly lower the computational costs of design.

5 Gradient Magnitude Model Selection

Gradient Magnitude Model Selection (GMMS) is a technique used in the Design Associate for selecting evaluation models in the context of a hillclimbing search procedure. The key idea behind this technique is illustrated by Figure 5. Suppose the system is running a hillclimbing algorithm to minimize $CourseTime$ as estimated by some approximate model. The values of $CourseTime$ returned by this approximate model are indicated by the curved line. Suppose further that the system is considering the hillclimbing step illustrated in the figure. If the error bars shown with solid lines reflect the uncertainty of the approximate model, the system can be sure that the proposed step will diminish the value of $CourseTime$. On the other hand, using the error bars shown with dotted lines, the system would be uncertain as to whether the true value of $CourseTime$ would improve after taking the proposed hillclimbing step. In the first case, the system could safely use the approximate model to decide whether to take the proposed hillclimbing step, while in the second case, the approximate model would not be safe to use for that decision. Thus GMMS evaluates the suitability of an approximate model by comparing error estimates to the magnitude of the change in the optimization criterion as measured by the approximate model.

GMMS actually operates in a manner that is slightly more general than outlined above. In particular, GMMS is implemented in the form of a function: $ModelSelect(p_1, p_2, K, M_1, ..., M_n)$. The parameters p_1 and p_2 represent artifacts under consideration during the design process (e.g., two different sailing yachts). The parameters $M_1, ..., M_n$ are an ordered list of the available models for evaluating artifact performance, where M_1 is the cheapest, and M_n is the most expensive. The *ModelSelect* routine returns the cheapest model that is sufficient for evaluating the following inequality:

$$M(p_1) - M(p_2) \geq K$$

Thus the selected model is sufficient for determining whether the performance of p_1 and p_2 differ by at least K. In order to evaluate forward progress in steepest-descent hillclimbing, as illustrated in Figure 5, the constant K is chosen to be zero. Our robustness-improving enhancements to steepest-descent hillclimbing occasionally require comparing artifacts using a non-zero tolerance level. In such cases, the *ModelSelect* routine takes a parameter K not equal to zero. GMMS can, in principle, be applied to any search algorithm that needs only to access the physical models in order to evaluate inequalities of the form shown above. Likewise, GMMS can in principle be applied to any of the modeling choices outlined in Figure 3.

6 Model Fitting and Error Estimation

We have experimented with GMMS using the choice of models for effective draft, T_{eff}, as a test case. Thus GMMS chooses between the algebraic approximate model and the $PMARC$ potential flow model described above. The accuracy of the algebraic approximation (relative to the $PMARC$ model) can be optimized by adjusting the value of the coefficient K. Our system fits the algebraic model and obtains an error estimate using the procedure outlined in Figure 6. The procedure takes as input two sets, A and B, of sample points in the space of candidate yacht designs. The set A is a small, sparsely distributed point set, while set B is a larger, more densely distributed point set. The system constructs two versions of the algebraic model, by choosing values for the fitting coefficient K. $Alg(A)$ is fitted against the "true" values from the sparse point set A. $Alg(B)$ is fitted against the "true" values from the dense point set B. In each case the "true" values are determined using the $PMARC$ as the "gold standard". Since $Alg(B)$ is fitted against the denser point set, this model is actually used during hillclimbing search; however, its error is estimated using $Alg(A)$, which was fitted against the sparser point set. In particular, the error in $Alg(B)$ is estimated by comparing $Alg(A)$ to $PMARC$ for all points in the set $B - A$. Two different error estimates result from this procedure: *Absolute-Error* is based on the assumption that errors in the algebraic model at nearby points in the design space are independent of each other.

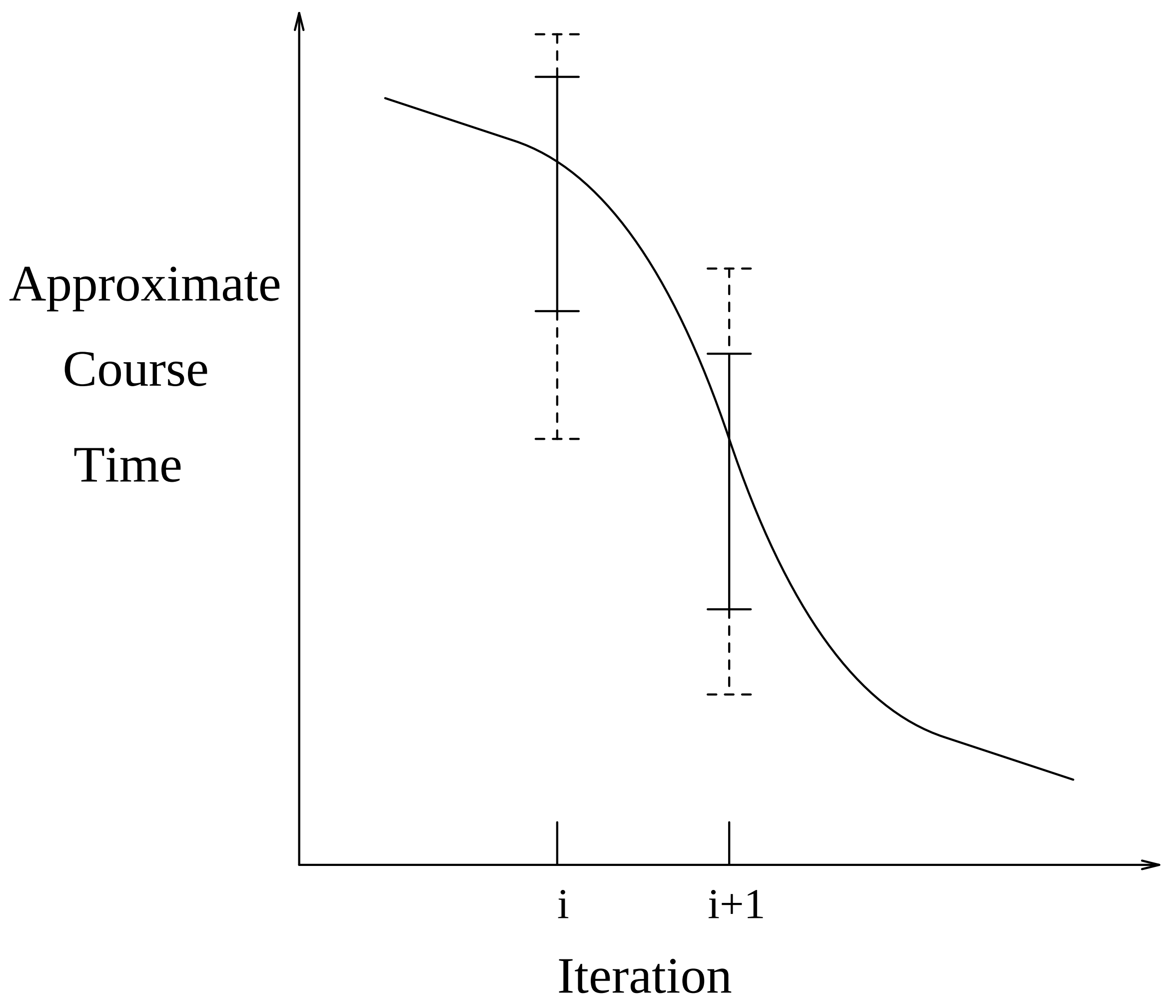

Figure 5: Gradient Magnitude Model Selection

Difference-Error takes into account the possibility that errors for nearby points may be correlated.

GMMS operates in a slightly different manner depending on which type of error estimate is available. Consider first how *Absolute-Error* estimates are used. Given two candidate yacht designs D_i and D_{i+1}, the system first evaluates the effective draft T_{eff} of each candidate using the algebraic approximation. The estimate of *Absolute-Error* is then used to find upper and lower bounds on the T_{eff} of each candidate. Each pair of bounds is then propagated through the rest of the velocity prediction program (Figure 2) to obtain an upper and lower bound on the $CourseTime$ of each candidate. If the $CourseTime$ intervals do not overlap, then the system knows that the step from D_i to D_{i+1} can be taken using the algebraic model. If the intervals do overlap, then the system must use $PMARC$ to obtain a better estimate of effective draft T_{eff} for each candidate.

When *Difference-Error* estimates are available, GMMS operates differently. After computing the effective draft of each candidate, the system considers two scenarios: (1) All of the *Difference-Error* occurs in the T_{eff} of D_i, and none occurs in the T_{eff} of D_{i+1}; (2) All of the *Difference-Error* occurs in the T_{eff} of D_{i+1}, and none occurs in the T_{eff}

1. Let A be a sparse point set in the design space $(u_1, \ldots, u_n)$.
   (a) Run $PMARC$ to find $T_{eff}(u)$ for each point in set A.
   (b) Fit coefficients in $Alg(A)$ to minimize average error over set A.
2. Let B be a dense point set in the design space $(u_1, \ldots, u_n)$.
   (a) Run $PMARC$ to find $T_{eff}(u)$ for each point in set B.
   (b) Fit coefficients of $Alg(B)$ to minimize average error over set B.
3. Estimate the error of $Alg(A)$ using the $PMARC$ as the "gold standard":
   - *Absolute-Error*$(Alg(A))$ = Average error in T_{eff} over all points in $B - A$.
   - *Difference-Error*$(Alg(A))$ = Average error in $|T_{eff}(u) - T_{eff}(u')|$ as function of $\Delta u_1, \ldots, \Delta u_n$, over all pairs (u, u') of points in $B - A$.

Figure 6: Error Estimation Technique

of D_i. In each case the system propagates the *Difference-Error* through the rest of the velocity prediction program, to obtain bounds on the $CourseTime$ of each candidate. The algebraic model is considered acceptable only if the $CourseTime$ intervals are disjoint under *both* scenarios. Similar methods can be used to apply Gradient Magnitude Model Selection to the other modeling choices shown in Figure 3.

Since GMMS is a heuristic method, it is not necessary that the error estimate be exact for each point in the design space. Overestimating the error will result in too little use of the approximate model, raising the cost of evaluation. Underestimating the error will result in overuse of the approximate model, leading the optimization along (possibly) less direct paths to the solution. Nevertheless, hillclimbing with GMMS should lead to a nearly optimal solution even when the approximate model is over-used. Recall that hillclimbing only terminates at local minimum points of the search space. Before stopping, the hillclimber usually encounters a region that is sufficiently flat to require use of the exact model in order to distinguish the performance of candidate designs. GMMS thus forces the hillclimber to switch to the expensive, but exact model in order to verify the presence of a local optimum.

The performance of GMMS can be enhanced by dynamically re-fitting the approximate model during the optimization process. This method proceeds from the observation that the optimal value of the fitting coefficient K in the T_{eff} formula will generally depend on the region of the design space in which the formula is applied. Suppose the algebraic model is periodically recalibrated during the search process, by adjusting this coefficient. The resulting approximate model will be more accurate in evaluation of designs near the latest recalibration point. It can therefore be used more often and provide greater savings over $PMARC$ than is possible with a fixed approximation.

We have implemented a "recalibrating" version of GMMS, "Recal-GMMS", to test out this strategy. Recal-GMMS operates as follows: Whenever $ModelSelect$ indicates that the current algebraic model cannot be used, the system runs $PMARC$ on the current design.

The computed value of T_{eff} is then used to recalibrate the coefficient of the algebraic model. Two different algebraic models are fit to the current region of the design space. In one model, the fitting coefficient K is treated as a constant. In the other model, K is expressed as a linear function of the parameters of the design space. This linear function is fit using $d+1$ *PMARC* evaluations for a d dimensional design parameter space. The required *PMARC* evaluations are obtained by selecting the $d+1$ most recent *PMARC* evaluations that yield a non-degenerate fitting problem. (Degeneracy is detected using a standard numerical singular value decomposition code.) In case no such non-degenerate set can be found, the system generates additional design parameter points that yield a non-degenerate set, and then evaluates them in order to carry out the fitting process. Of the two recalibrated algebraic models, the linear model is actually used to compute effective draft T_{eff}. The error of this linear model is estimated to be the absolute value of the difference between the linear model and the constant model.

7 Experimental Results

We have tested our approach to model selection in a series of experiments comparing various model selection strategies. In particular, we investigated the five model selection strategies listed in Figure 7. The five strategies were each run on four separate design optimization problems. The problems differed in both the initial yacht prototypes, and in the yacht design goals. Two shape modification operators were used for the optimizations, i.e., `Scale-Keel`, which changes the depth of the keel, and `Invert-Keel`, which alters the ratio between the lengths of the top and bottom edges of the keel. The results of these runs are summarized by the table in Figure 8. For each model selection strategy, the table gives a measure of the quality of the final design, and a measure of the computational cost needed to find the final design, each averaged over all four test problems. The quality of a design D is measured as the difference between the $CourseTime$ of D and the $CourseTime$ of the "optimal" design, i.e. the best design found by any of the five strategies. The computational cost of finding a design is measured by counting the number of *PMARC* evaluations needed to carry out the design optimization, since *PMARC* is by far the most expensive part of the design process.

The results in Figure 8 illustrate a tradeoff between computational cost and the quality of the optimization. In terms of computational expense, measured by the number of *PMARC* evaluations, the strategies can be ranked in the order shown, with "Alg-Only" being the cheapest and "PMARC-Only" being the most expensive. "Alg-Only" comes out being the cheapest because it never invokes the *PMARC* potential flow code. "PMARC-Only" is the most expensive because it always invokes the *PMARC* potential flow code. Notice that each of the three non-trivial model selection strategies ("Recal-GMMS", "GMMS" and "CTO") is cheaper than the "PMARC-Only" strategy. Each avoids some of the *PMARC* runs that occur under the "PMARC-Only" strategy. In fact , the computationally cheapest of the three, "Recal-GMMS", incurs only about 59% of the computational expense of the "PMARC-Only" strategy. Notice that the "Alg-Only" strategy yields yacht designs of lower quality than those produced using the other strategies. Lower quality designs are obtained because the algebraic model causes the hillclimber to terminate at a point that is not a local optimum in terms of the more accurate "*PMARC*" model. In contrast to this, all of the

- **Alg-Only:** Only the algebraic model is used for evaluation of effective draft.
- **PMARC-Only:** Only the *PMARC* potential flow code is used for evaluation of effective draft.
- **GMMS:** Gradient magnitude model selection is used to select between the algebraic and *PMARC* models for effective draft. Errors are estimated using the *Difference-Error* formula. Errors are propagated through *VPP* using the difference error propagation method.
- **Recal-GMMS:** GMMS, with the addition that the algebraic model is recalibrated according to *PMARC* data collected during the optimization. Errors are estimated by comparing locally fit constant and linear models. Errors are propagated through *VPP* using the absolute error propagation method.
- **CTO (Cheap-to-Optimal):** Only the algebraic model is used until an initial optimum is reached. Then only the *PMARC* model is used until a final optimum is reached.

Figure 7: Model Selection Strategies

other four strategies achieve the same quality levels. Higher quality designs are obtained because these strategies cause the hillclimber to terminate at points that really are locally optimal in terms of the *PMARC* model.

Strategy	Compute Cost (*PMARC* Evals)	Design Quality (Lag in Seconds)
`Alg-Only`	0.00	-351
`Recal-GMMS`	152.00	0
`CTO`	207.75	0
`GMMS`	250.00	0
`PMARC-Only`	257.50	0

Figure 8: Comparison of Model Selection Strategies

The "Alg-Only" and "Recal-GMMS" strategies are Pareto optimal. Neither of these two strategies is dominated in quality and computation cost by any of the other three strategies. In contrast, none of the other three strategies ("PMARC", "GMMS" and "CTO") is Pareto optimal. Each is dominated by the "Recal-GMMS" strategy, since the "Recal-GMMS" strategy achieves the same quality as each at a lower computational cost. In order to choose between "Alg-Only" and "Recal-GMMS" one must supply some criterion for balancing the quality of a design against the amount of computation cost expended during the design process. In the yacht design domain, the choice is fairly easy. America's Cup yacht races are often won and lost by a few seconds. Considerations of quality therefore tend to outweigh considerations of computation cost. In this application domain, our results indicate that "Recal-GMMS" is the best model selection strategy.

8 Ongoing Research

Ongoing research is aimed at applying our GMMS techniques to other model selection choices that arise during hillclimbing search in the yacht design domain, as described in Figure 3. We are especially interested in using GMMS to decide when to reuse prior evaluation results, and when to use linear approximation models. These two types of approximation are very general and can be applied to a wide variety of design problems. If GMMS can be shown useful for these decisions, it will be established as a widely applicable model selection technique. We also plan to test our GMMS techniques in domains other than yacht design.

Longer term research is aimed at investigating model selection problems that arise in parts of the design process other than hillclimbing search. Models of physical systems can be used to support computer-aided design in a variety of ways other than direct evaluation of candidate designs. For example, physical models can be used in sensitivity analyses that enable engineers to decide which design parameters to include in the search space. Each design task that depends on a physical model will lead to a distinct model selection problem. We are therefore attempting to classify the modeling tasks that arise in computer-aided design and to develop model selection methods for each of them.

9 Acknowledgments

This research was supported by the Defense Advanced Research Projects Agency (DARPA) and the National Aeronautics and Space Administration (NASA). (DARPA-funded NASA grant NAG 2-645.) It has benefited from discussions with Saul Amarel, Martin Fritts, Andrew Gelsey, Haym Hirsh, John Letcher, Chun Liew, Ringo Ling, Gerry Richter, Nils Salvesen, Louis Steinberg, Chris Tong, Tim Weinrich, and Ke-Thia Yao.

References

[Addanki *et al.*, 1991] S. Addanki, R. Cremonini, and J. Scot. Graphs of models. *Artificial Intelligence*, 50, 1991.

[Cerbone and Dietterich, 1991] G. Cerbone and T. Dietterich. Knowledge compilation to speed up numerical optimization. In *Proceedings of the Eighth International Workshop on Machine Learning*, Evanston, IL, 1991.

[Falkenhainer and Forbus, 1991] B. Falkenhainer and K. Forbus. Compositional modeling: Finding the right model for the job. *Artificial Intelligence*, 50, 1991.

[Falkenhainer, 1992] B. Falkenhainer. A look at idealization. Working Notes of the AAAI Workshop on Approximation and Abstraction of Computational Theories, San Jose, CA, 1992.

[Letcher *et al.*, 1987] J. Letcher, J. Marshall, J. Oliver, and N. Salvesen. Stars and stripes. *Scientific American*, 257(2), August 1987.

[Ling and Steinberg, 1992] R. Ling and L. Steinberg. Model generation from physical principles: A progress report. Technical Report CAP-TR-9, Department of Computer Science, Rutgers University, 1992.

[Press *et al.*, 1986] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes.* Cambridge University Press, New York, NY, 1986.

[Tcheng *et al.*, 1991] D. Tcheng, B. Lambert, S. Lu, and L. Rendell. Aims: An adaptive interactive modeling system for supporting engineering decision making. In *Proceedings of the Eighth International Workshop on Machine Learning*, Evanston, IL, 1991.

[Weld and Addanki, 1991] D. Weld and S. Addanki. Query-directed approximation. Technical Report 90-12-02, Department of Computer Science and Engineering, University of Washington, Seattle, WA, 1991.

[Weld, 1991] D. Weld. Reasoning about model accuracy. Technical Report 91-05-02, Department of Computer Science and Engineering, University of Washington, Seattle, WA, 1991.