Pursuing the Pareto Paradigm: Tournaments, Algorithm Variations & Ordinal Optimization

Mark Kotanchek[1], Guido Smits[2], and Ekaterina Vladislavleva[3]

[1] Evolved Analytics Inc. mark@evolved-analytics.com
[2] Dow Benelux B.V. guido@evolved-analytics.com
[3] Tilburg University katya@evolved-analytics.com

The **ParetoGP** algorithm which adopts a multi-objective optimization approach to balancing expression complexity and accuracy has proven to have significant impact on symbolic regression of industrial data due to its improvement in speed and quality of model development as well as user model selection, [1], [2], [3]. In this chapter, we explore a range of topics related to exploiting the Pareto paradigm. First we describe and explore the strengths and weaknesses of the **ClassicGP** and **ParetoGP** variants for symbolic regression as well as touch on related algorithms. Next, we show a derivation for the selection intensity of tournament selection with multiple winners (albeit, in a single-objective case). We then extend classical tournament and elite selection strategies into a multi-objective framework which allows classical GP schemes to be readily Pareto-aware. Finally, we introduce the latest extension of the Pareto paradigm which is the melding with ordinal optimization. It appears that ordinal optimization will provide a theoretical foundation to guide algorithm design. Application of these insights has already produced at least a four-fold improvement in the **ParetoGP** performance for a suite of test problems.

1 Introduction

The **ParetoGP** algorithm, [1], was originally inspired by the need to sort through the plethora of results produced by application of genetic programming to symbolic regression of industrial datasets. Once the key insight occurred that the expressions of interest would lie along the Pareto front trading off expression accuracy and expression complexity (which was assumed to be a metric linked to the risk of overfitting), it was a natural evolution to modify the genetic programming algorithms to accommodate our view that the Pareto front is where the interesting models resided which should be explored during the evolutionary process. The resulting **ParetoGP** algorithm was interesting on a number of fronts. First, the orders-of-magnitude improvement

in modeling efficiency opened up the size and scope of data sets which could be handled with the natural variable selection capability proving to be an important additional benefit for complementary nonlinear modeling techniques such as neural networks and support vector regression. The second aspect was that the user was presented with a natural set of models which explored the trade-off between expression complexity and accuracy and, thereby, facilitated post-processing model selection for subsequent model exploitation. The third aspect was that **ParetoGP** required significantly smaller population sizes for evolving good solutions [3] than conventional GP theory predicted. We believe that this is due to the inclusion of an archive. Finally, the industrial impact of the resulting modeling successes inspired additional research into the algorithm [3], its applications, [4] and additional extensions and enhancements [5].

In this chapter we briefly review the **ClassicGP** and **ParetoGP** algorithms and their characteristics and typical parameter settings. With that context established, we take a look at tournament selection within the context of multiple objective optimization and propose a **ParetoTourneySelect** method which facilitates a Pareto-aware implementation of the classic GP methodology. This method is attractive due to its simplicity and ease of tuning the selectivity focusing. Prior to introducing the **ParetoTourneySelect** method, however, we review the selection intensity of single-objective tournament selection with single and multiple winners. This is of practical importance since single-objective tournament selection is often used within the **ParetoGP** algorithm as its selection method.

Finally, we introduce the notions of ordinal optimization and its application to genetic programming algorithm design. Although still in the early stages of the research, the insights derived from these concepts have been applied to **ParetoGP** to produce significant improvements in both modeling efficiency and consistency as measured by the shape of the resulting Pareto fronts.

2 Pareto-Aware GP - Variations on the Pareto Theme

A number of researchers independently explored parallel optimization of competing objectives in genetic programming. We partition their approaches into three broad algorithmic categories: ClassicGP, ParetoGP and Hierarchal Fair Competition (HFC). In this section, we briefly review the ClassicGP and ParetoGP classes.

When evolutionary search uses competing criteria, the optimal solution may not exist. Instead,a set of alternatives,called the Pareto-optimal set,will be an optimum. Pareto-optimal set consists of individuals for which no other individual is superior in all criteria. In the objective space (e.g.model error vs.model complexity) this set will form the Pareto front. So, each member of the Pareto-optimal set surpasses all individuals of the search space (or all

considered so far) in at least one objective, and hence becomes a candidate for careful consideration and protection.

2.1 Variations on the Pareto Theme

Within the **ClassicGP** framework, Bleuler, et al, [6], assigned breeding rights based upon the SPEA2 metric of a population with members of the Pareto front persisting across generational boundaries. Saetrom and Hetland, [7], also essentially followed this approach. De Jong & Pollack, [8], proposed a Pareto front-centric approach wherein they synthesize new models each generation which are merged with a persistent Pareto front and used to define an updated Pareto front with propagation restricted to those models on the front (this approach did not work very well).

Recently, we have recognized that **HFC**, [9], should be included in the Pareto-aware algorithm category since it partitions models based upon accuracy fitness and restricted breeding and competition to models operating on similar levels of the fitness axis. Although not explicitly using the Pareto front by name, this approach is functionally Pareto-aware.

Obviously, the authors and their colleagues have pursued the **ParetoGP** variant [1], [2]. However, the development of the **ParetoTourneySelect** strategy has prompted them to include the **ClassicGP** framework within their research repertoire.

Strangely, the publications from the other researchers have not noted the explosive improvement in computational efficiency and robustness which has been associated with **ParetoGP** on real-world problems. (The exception being **HFC** which Erik Goodman noted had a similar improvement in efficiency in private conversation.) This may be due to the nature of their test suites or the computational loads of the multi-objective selection schemes used to assign breeding rights.

2.2 ClassicGP & ParetoGP

In the **ClassicGP** approach, illustrated in Figure 1, starting from a supplied model set (if not supplied, random models will be synthesized), populations are supplied to each of the parallel and independent evolutionary runs. Each run consists of n generations wherein survival-of-the-fittest is applied to assign breeding rights for the next generation. We enhanced the classic GP approach by partitioning each evolution into **cascades** - groups of parallel and independent short evolutions (**runs**) of different populations to prevent inbreeding and maintain the diversity of solutions. Results consolidated at the end of a cascade may be used to seed subsequent cascades. At the end of the processing, the reported models are selected from the final cascade results.

Conversely, as illustrated in Figure 2, the **ParetoGP** approach uses a new entity, called archive, co-existing with a population. The archive survives

Fig. 1. ClassicGP Modeling Flow - There is room for Pareto-aware selection strategies such as **ParetoTourneySelect**. Notice, the cascades are executed in sequence and runs - in parallel

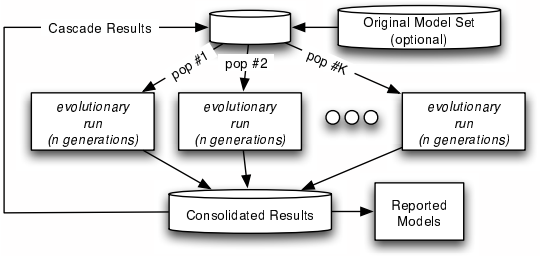
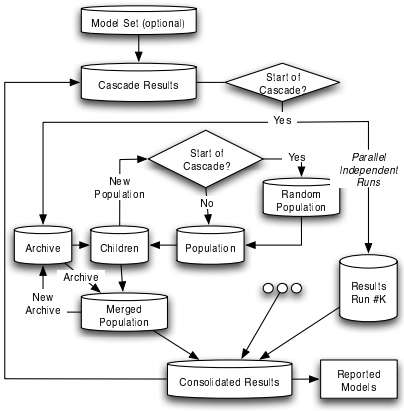


Fig. 2. ParetoGP Modeling Flow - Because **ParetoGP** defines its archive using Pareto layers from the evolving populations, it is intrinsically a Pareto-aware GP algorithm even if conventional single-objective strategies are used to select from the archive and population



across the cascade boundaries, while the population for each parallel independent run (however, only one run is typically run in parallel) is wiped out and replaced by a new random population at the start of a new cascade. As shown in Figure 2, breeding is pairwise with one parent from the archive and the other from the population. At the end of a generation, after a new population is created, the archive is updated with the Pareto layers from the archive combined with the new population until the specified archive size is met. Because the archive is defined and updated using Pareto layers, **ParetoGP** is intrinsically a multi-objective algorithm even if more classical selection methods such as single-objective tournament selection are used to select the breeding pool from the population and archive.

Choosing model complexity as a second optimization criterion in symbolic regression involves a trade-off between exploration for good structural foun-

dations and exploitation of those foundations to achieve models which are both parsimonious and accurate. In **ClassicGP**, the exploration is accomplished by the parallel independent evolutionary runs with the exploitation provided by the subsequent cascades as the foundation structures are refined and recombined to produce the desired quality models. As a result, generally the proper balance is to have many parallel independent runs feeding relatively few cascades. In contrast, for the **ParetoGP** approach, the exploration comes in primarily through the random populations introduced with each cascade whereas the exploitation comes from the persistence of the archive which survives across the evolution boundaries. Since new genetic material is introduced with each random population, the exploitation continues despite the maturation of the archive solution. Hence, for **ParetoGP**, many cascades and fewer independent parallel runs is generally required. However, despite the similarity of the resulting Pareto front results, the founder effect still applies so that apparent structures from a ParetoGP thread will generally differ across independent evolutions.

We recommend extending the ClassicGP strategy from the conventional single-objective realm (wherein accuracy is reduced by a complexity penalty) into the multi-objective by using multiple selection criteria (e.g., **ParetoTourneySelect** or **ParetoEliteSelect**). In the next sections we will show that this is a very powerful extension.

It appears that defining a methuselah function of the top 30% of the population mimics the effect of the ParetoGP archive and allows ClassicGP to be competitive with ParetoGP performance. It may be, however, that the inflicted influx of new genetic material at cascade boundaries may be advantageous in simultaneously maintaining exploration along with exploitation.

3 Tournament Selection Intensity - Single & Multiple Winners with One Objective

In the GP realm, tournament selection appears to have dominated its competition (proportional, rank-based, elitist and random selection) due to being efficient and able to balance exploration and exploitation simply by tuning the tournament size used to select a winner. One reason for tournament selection being efficient as well as robust is that the ranking used to identify the winner is *ordinal* rather than depending on the absolute fitness values relative to the rest of the population. This helps to avoid premature convergence in selecting from a population as well as being computationally easy. We shall revisit the implications of ordinality later in the chapter.

The likelihood of being both selected for a tournament as well as emerging from the tournament as a winner is known as the selection intensity. In this section we explore the selection intensity as a function of population size, n, tournament size, t, and number of winners, w. As we shall see, allowing multiple winners to emerge from each tournament pool adds an additional ability

to shape the selection intensity. There are two basic tournament selection variants depending upon whether we allow replacement or not. If we allow replacement, a model can compete against itself for breeding rights. While not physically realizable, this is often easier algorithmically than ensuring that the tournament pool competitors are unique when we are making a random draw. From a practical perspective, the selection intensities with or without replacement are comparable for reasonably large population sizes. With that as an introduction, we turn our attention first to the tournament selection with replacement.

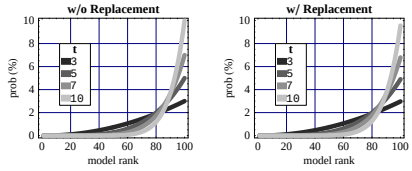
3.1 Tournament Selection with Replacement - single winner situation

This approach is courtesy a conversation with Steffen Christensen. Basically, we make a geometric argument that if we make t draws from a population - and allow replacement - then we are really defining a location in a t-dimensional space wherein each dimension is quantized. In order to get into the game, the individual must be selected. In order to win the tournament, no better individual may be selected. To compute the likelihood of being a winner, we calculate the probability that the selected ensemble (where any given ensemble is represented by a node, i.e., quantized location in the t-dimensional space) doesn't contain any higher quality individuals, this is simply represented by the volume of the hypercube which excludes the higher-ranked models divided by the hyper-volume of the overall space which includes the entire population. This is calculated as $\frac{r^t}{n^t}$ where r is the rank of the individual in question (with larger numbers corresponding to higher rank). Excluding the likelihood of only selecting individuals from the lower ranked models corresponds to identifying the probability that only those models are selected, i.e., $\frac{(r-1)^t}{n^t}$. Combining these results gives us the probability (a.k.a., *selection intensity*), p, of an individual with rank r winning a randomly selected tournament of size t from a population of n individuals.

$$p = \frac{r^t - (r-1)^t}{n^t} \tag{1}$$

In Figure 3, we show the flexibility possible simply by varying the tournament size. Note the nonlinear nature of the selection intensity as the tournament size increases. Also note that due to the possibility of replacement, that the likelihood of the top-ranked individual being selected is slightly *less* than it would be in the selection without replacement cases. For example, with a population size of 100 and a tourney size of 10, the top individual will win breeding rights 9.56% of the time whereas without replacement, it should achieve 10% of the breeding rights.

Fig. 3. Selection Intensity Behavior for a Single Winner. Selection intensity for tournaments of different sizes with a single winner for both the tournament selection with and without replacement. t stands for the size of the tournament



3.2 Selection Intensity without Replacement - single winner situation

Now let us turn our attention to the situation wherein each member of the tournament pool must be unique. Under the assumption of random selection, the likelihood of any given individual being selected is simply the ratio of the tournament size to the population size, i.e., $\frac{t}{n}$. To win the tournament, we have the restriction that none of the other selections can be higher ranking than the r^{th} individual. The likelihood of this happening - conditioned on the r^{th} individual already having been selected is the product of the successive likelihood of not selecting a better individual in any of the remaining t-1 draws to fill the tournament

$$\frac{r-1}{n-1}\frac{r-2}{n-2}\cdots\frac{r-(t-1)}{n-(t-1)} \tag{2}$$

Notice that the pool decreases with each selection due to our assumption of unique individuals being drawn. Also note that we need to handle the special case when we have negative numbers; this happens at the point where there is no chance that the individual will win a tournament - e.g., the bottom t-1 individuals. Pulling this together, we have the result,

$$p = \begin{cases} \frac{t}{n}\prod_{k=1}^{t-1}\frac{r-k}{n-k} & r \geq t-1 \\ 0 & r < t-1 \end{cases} \tag{3}$$

This response behavior is also shown in Figure 3. As noted previously, despite the visual similarity of the two plots, there is a slight difference due to the avoidance of repeated models in a given draw.

3.3 Selection Intensity without Replacement - multiple winner situation

If we have more than one winner in a tournament, it is a fairly simple to extend the above logic. Assuming a tournament size, t, which has w winners, we need to consider w scenarios ranging from the situation wherein the given entity is the top ranked tourney contender to the situation where it is the w^{th}ranked

member of the pool (and, therefore, barely squeaking into breeding status). Under the top-ranked member scenario, we simply have the prior single winner situation, i.e.,

$$\frac{r-1}{n-1}\frac{r-2}{n-2}\cdots\frac{r-(t-1)}{n-(t-1)} \tag{4}$$

Note that the above is the product of t-1 terms since we are implicitly assuming the given ranked model has entered the pool with a probability of $\frac{t}{n}$ where n is the population size. The probability of there being one higher-ranked model is

$$\binom{t-1}{1}\left(\frac{r-1}{n-1}\frac{r-2}{n-2}\cdots\frac{r-(t-2)}{n-(t-2)}\right)\left(\frac{n-r}{n-(t-1)}\right). \tag{5}$$

Here we need the binomial coefficient, $\binom{t-1}{1}$, to account for the fact that there are t-1 different ways that the better model can enter the tourney - under the conditional assumption that the r^{th} model has been selected. If the top ranked model was being examined, then n-r would be zero so the series would naturally truncate and zero out all successive scenarios. In a similar vein, the probability of two higher ranked models is

$$\binom{t-1}{2}\left(\frac{r-1}{n-1}\frac{r-2}{n-2}\cdots\frac{r-(t-3)}{n-(t-3)}\right)\left(\frac{n-r}{n-(t-1)}\cdots\frac{n-r-1}{n-(t-2)}\right). \tag{6}$$

This sequence of situations to be considered would terminate at the point wherein there are w-1 higher ranked models in the pool, i.e.,

$$\binom{t-1}{w-1}\left(\frac{r-1}{n-1}\cdots\frac{r-(t-w)}{n-(t-w)}\right)\left(\frac{n-r}{n-(t-1)}\cdots\frac{n-r-(w-2)}{n-(t-w-1)}\right). \tag{7}$$

Assembling the terms from the possible scenarios lead us to the summary expression for the probability,

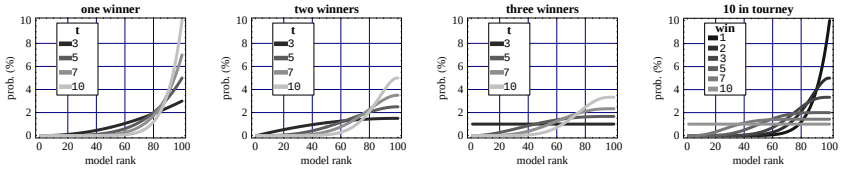
$$p=\sum_{s=1}^{w-1}\left(\binom{t-1}{s}\left(\prod_{k=1}^{t-(s+1)}\frac{r-k}{n-k}\right)\left(\prod_{g=1}^{s}\frac{n-r-(g-1)}{n-(t-g)}\right)\right)\cdot\ldots$$
$$\ldots\cdot\frac{t}{n}\cdot\prod_{k=1}^{t-1}\frac{r-k}{n-k}+\frac{t}{n}\left(\prod_{k=1}^{t-1}\frac{r-k}{n-k}\right)^2, \tag{8}$$

Note that the above expression is the probability of being selected for and winning a given tournament. Under normal circumstances where we are trying to assemble a group of breeders, we would need to execute fewer tournaments

if we had multiple winners. Hence the above probability would typically be normalized by the number of winners to get the population percentage.

In Figure 4 we show the effect of varying the number of winners and tournament sizes. The important thing here is the ability to shape the selectivity. A perusal of the GP literature seems to show that a tournament size of between three and five is generally used with a single winner emerging from each tourney.

Fig. 4. Multiple Winners in Tournaments. The selection intensity effect of changing the number of winners from a tournament for different tournament sizes for tournament selection without replacement. Views varying the tournament size as well as holding the tourney size constant and varying the number of winners are shown



3.4 Tournament Selection Intensity - Summary

In the above we have developed expressions for the tournament selection intensity as a function of population size, tournament size and number of winners. This allows us to have an explicit understanding of the implications of parameter settings as well as the ability to shape the selection intensity to produce elitist-like behavior by having multiple winners.

4 Tunable Pareto-Aware Selection Strategies

As noted previously, the tournaments have emerged as the dominant selection strategy in genetic programming because of their simplicity, robustness and effectiveness. Intuitively, we would like to incorporate this strategy in the Pareto-aware GP implementations; unfortunately, the Pareto paradigm implies multiple definitions of success so we have the problem of declaring the winner from a tournament.

There are a wide variety of selection strategies which have been used in multi-objective-optimization (MOO): NGSA, NSGA-II, SPEA, SPEA2, MOGA, NPGA, DMOEA, VLSI-GA, PDE, PAES, PESA, etc. Most of these rely upon the notion of dominance (the number of models which a given model

beats), domination (the number of models which beat the given model), dominance layer or a combination with breeding rights awarded based upon the scores relative to the entire rest of the population. There are three fundamental problems with most of these selection strategies:
1) **selection effort** - it is computationally intensive to evaluate the population for large population sizes (and the curse-of-dimensionality means that we want large population sizes in multi-objective-optimization),
2) **requirement for global awareness** - the requirement for global awareness of the population and their relative fitness makes it difficult for the selection methods to scale well with population size or number of objectives,
3) **tunability** - we want to be able to have an easily controlled parameter which will tweak the exploitation vs. exploration balance in selecting models from the population for development.

Our approach to implementing a multi-objective tournament selection strategy (which we call **ParetoTourneySelect**) is very simple: we form pools of randomly selected models and, since we cannot distinguish between them, the winners are all of the models on the Pareto front of that pool. We keep repeating this process until we achieve the desired selection size. The attraction of this strategy is that identifying the Pareto front of a population is significantly easier computationally than doing the ranking associated with conventional schemes. Additionally, working with smaller subsets of the population improves the efficiency of the Pareto front identification. Of course, there is an additional obvious selection strategy - **ParetoEliteSelect** - in which we assemble Pareto layers from the total population until a specified elite size is achieved.

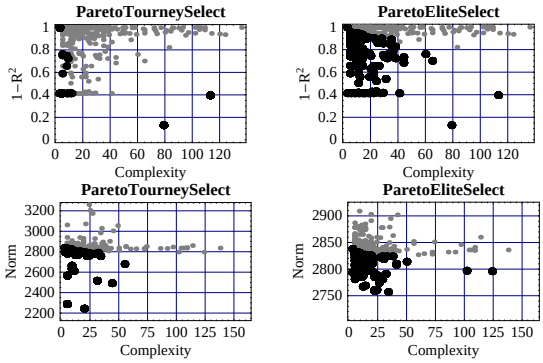
In this section we will use a very simple response model without any additive noise, $x_1^2 (1 + x_2)$.

One hundred random evaluation points are created using four variables each distributed over a range of [-10,10] with the response function responding to the first two and the remaining two being spurious. We also synthesize 1000 random symbolic regression models with unique genomes (albeit, not necessarily unique phenotypes) and evaluate the models against the observed input-output data using a variety of accuracy metrics (2-norm and 1-R^2) as well as a model complexity metric (total sum of the sum of the nodes of all subtrees in the genome).

In Figure 5, we show the results from identifying 1000 models from the population using a **TournamentSize** or an **EliteSize** of 10% of the population size, as appropriate. Note that this method has focused the selection process on the best models from a multi-objective perspective. As we can see, both selection strategies focus on models which are candidates for further development.

In Figure 6 we look at the distribution of the selected models and we see the focusing effect of the Pareto tournament selection approach. Cleaning the population and removing the models with duplicate phenotypes (defined as having identical fitness values) improves the diversity in model selection and

Fig. 5. Selection behavior of the ParetoTourneySelect and ParetoEliteSelect strategies - Here we select 1000 models from a population of 1000 random models with no duplicate phenotypes with an elite size of 10% and a tourney size of 10% of the population, as appropriate. We ran evolutions with model selection in two accuracy metrics and plotted models that got selected in black. Note the strong focusing of the **ParetoTourneySelect** strategy favoring models from an interesting area of the objective space (here, an area of low model error and low complexity)



strengthens the focusing effect of the **ParetoTourneySelect** strategy. Note that less than 100 of the 1000 model population would have been selected for further evolutionary exploration and this subset is strongly skewed towards a relative handful of models. However, this focusing considers the multiple objectives. For comparison, we also show the results from the **ParetoEliteSelect** strategy.

One of our stated goals at the start of this section was to provide a multi-objective selection function which replicated the one-objective characteristics of tournament selection for ordinality, localized comparison and ease of selection intensity tuning. We have accomplished the first two; before we leave this topic, let us demonstrate the tunability capability in Figure 7.

Selection Efficiency & Discussion

Table 1 and Table 2 show tabular looks at two scenarios. The first shows the effect of tournament size on computation time as well as model diversity resulting from applying **ParetoTourneySelect** to the 1000 model population (evaluated with the norm metric). The second shows the effect of different population sizes holding the tournament size fixed as fraction of population size. Note that the selection diversity (as measured by the % of population selected) seems to be tied more to tournament size than to fraction of population used within the tournament - which is an interesting result to be investigated in the future. Also note that the time per model selected increases with increasing

Fig. 6. Metric vs. Pareto Tournament & Elite Selection - Selection intensity of the **ParetoTourneySelect** as a function of accuracy metric for 1,000 selections from 1,000 models with a 10% tournament size or a 10% elite size. Note the strong focusing of the **ParetoTourneySelect** strategy (depicted in black) relative to the flat **ParetoEliteSelect** strategy (depicted in gray)

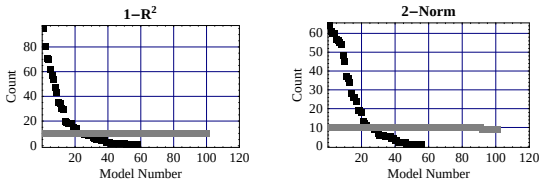
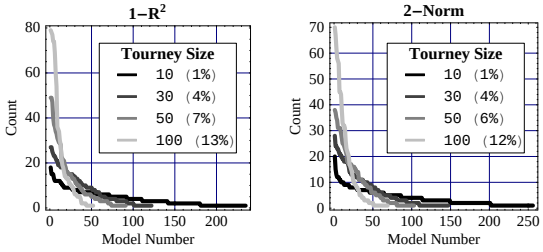


Fig. 7. Tourney Size vs. Selection Intensity - Here we show the tuning effect for selecting 1000 models from the unique phenotype random models in the test case. Note that although the choice of fitness metric has an effect on the selectivity, the shape of the selectivity is controlled by changing the tournament size



tournament size since a lower fraction of the tournament emerge as winners as the tournament size increases. A surprising result here is that evolutions with the the **ParetoTourneySelect** method are executed in 33% less time than those using the classical accuracy-based tournament selection scheme! The fact that it happens despite all overhead in the Pareto front calculation for **ParetoTourneySelect** is again explained by focusing of the multi-objective tournaments on a small fraction of 'potentially' optimal individuals and keeping the average size of expressions smaller, which makes for faster evolutions. We illustrate these results in Figure 8.

5 Ordinal Optimization & Application to Symbolic Regression

The basic notion of ordinal optimization (OO) [10] is that for computationally hard problems, our target is generally a *good enough.* solution rather than the true optimal model. Since this is a similar viewpoint to industrial application of symbolic regression, an exploration of the ordinal optimization and its

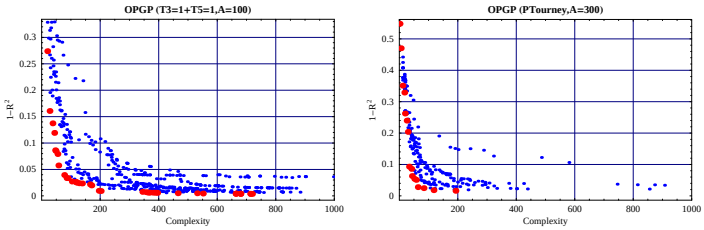
Table 1. Selection Time and Diversity vs Tournament Size - Here we look at the effect of tournament size on both the model diversity and selection time. The Pareto tournaments was applied to the norm-metric set of 1000 models (processed so that there were no more than two of any given phenotype) and 1,000 models were selected. As a reference point, calculating the model fitness requires 1.7 seconds for the population.

Pop Size	Tourney Size	% Selected	Time (sec)	Time/Model (ms)
1000	10	24	2.44	2.44
1000	20	18	2.31	2.31
1000	40	11	2.45	2.45
1000	80	6	2.75	2.75
1000	100	5	2.83	2.83

Table 2. Selection Diversity and Time vs. Population Size - Here we assume that the population size is maintained and that the tournaments consist of 10% of the population so, for example, 100 models are selected from the 100 model population using a sequence of 10 model (10% of population) tournaments. (Subsets of the model set of Table 2 are used.) Looking at the time/model selected column shows the increasing Pareto front identification effort required with increasing tournament sizes. Also note that the model fitness evaluation time scales linearly so the time required for the 100 model population for this example is 0.17 seconds - which makes the selection effort relatively small for that size population.

Pop Size	Tourney Size	% Selected	Time (sec)	Time/Model (ms)
100	10	34	0.03	0.3
200	20	20	0.1	0.52
400	40	13	0.34	0.85
800	80	7	1.1	1.38
1000	100	4	1.68	1.68

Fig. 8. Comparison of classical tournament selection with the ParetoTourney. We plot the results of 11 independent evolutions per each method, and overall Pareto-optimal sets (depicted in bold black). Note that the Pareto tournaments (plot at the right) focus the energy on the most interesting area of the objective space. Surprisingly, while computationally demanding, evolutions with the **ParetoTourneySelect** method have shown 35% improvement in the execution time compared with the classical single-objective tournament selection



application to symbolic regression is warranted. The foundation principles of OO are first, that it is easier to get a ranking of candidate solutions than it is to exactly compute the fitness values of the candidates. As a result, if we can quickly identify a subset of solutions which are worth further investigation, then we should be able to improve the efficiency of the model search. Second, goal softening helps to smooth and direct the search i.e. instead of asking for the absolute best it is better to ask for good enough with high probability.

5.1 Concepts & Tuning Parameters.

The OO mantra of "goal softening" could be expressed in the management literature as "fail forward" - in other words, identify promising solutions as quickly as possible and then pursue them. To a large extent, GP is already ordinal in nature; however, as we shall see, the OO viewpoint leads to some additional gains. First, let us review the aspects of Pareto-aware GP which we can control to allow us to fail forward:

- **fitness evaluation** - evaluating the model quality is, typically, where most of the computational effort is spent in symbolic regression. Within that broad category, there are two knobs to turn:
  1) **fitness metric** - even for symbolic regression choice of fitness metric has a computational load component. Similarly, model complexity could be represented by a variety of schemes as simple as node count or as complex as nonlinearity estimates. This aspect is even more significant in other genetic programming applications since a first-principles model evaluation will require much more effort than needed for a first-order approximation.
  2) **data subset size** - rather than evaluating the model at all data points, subsets of the data can be used with, generally, a linear shift in the computational load required. If the data subset is static, then care must be taken to properly balance the subset so that it is representative of the overall data set. If the subset varies, then care must be taken that apples-to-apples comparisons are made of quality results.
- **variable selection** - one of the best features of the Pareto-aware symbolic regression is the automatic variable selection during the evolutionary process so that ill-conditioned data sets with a plethora of nuisance variables may still be effectively analyzed. However, if the spurious variables can be rejected, the efficiency of the modeling can be increased.
- **selection method** - selection has two aspects from a computational efficiency viewpoint. The first is the effort required to identify quality solutions (which is a strength of the single-objective tournament selection). The second aspect is the focusing efficiency and controllability. For multi-objective optimization, this is a strength of the Pareto tournaments since it has a fuzzy threshold to separate good and bad models which can easily be tuned. Also note that for multiple objectives that criteria subsets may be used for model selection analogous to the data subsetting.

- **population (& archive) size** - the size of the model set, obviously, has a direct mapping into the computational load. One attraction of OO is that it may provide a theoretical foundation for identifying the proper problem-specific population size.
- **generations per run** - typically, the number of generations in a run corresponds to the exploitation effort of discovered solutions. In the spirit of OO we would want to use minimal generations in the early cascades with increased generations in the later stages as we refine and explore the foundation models.
- **runs per cascade** - the number of parallel runs within a cascade corresponds to the exploration component of the symbolic regression. Especially for a **ClassicGP** approach, OO would seem to guide us towards many short runs in the early cascades and shifting towards fewer longer runs in the latter stages.
- **cascades per evolution** - the number of cascades determines the extent of model exploitation. For **ParetoGP** the inclusion of additional cascades is generally worthwhile due to the influx of new genetic material at the cascade boundaries; for **ClassicGP**, additional cascades represent diminishing returns after a certain point since the exploration component is primarily associated with mutation.
- **number of independent evolutions** - the founder effect results in early fit solutions dominating the population. Therefore, consistency of the functional quality resulting from independent evolutions is an indicator that an appropriate evolutionary effort has been applied.

5.2 Initial Application of OO to ParetoGP

Smits & Vladislavleva [5] adopted the viewpoint that the majority of symbolic regression time is spent in the fitness evaluation and, therefore, performing an ordinal evaluation using random subsets of the data rather than the complete set was an attractive starting point for exploiting OO within GP. They looked at three cases using **ParetoGP** wherein they varied the characteristics of the cascades within the evolution process. The quality metric to compare the results was the area under the modeling Pareto front for a 1-R^2 accuracy metric and a genome complexity metric ranging from 1 to 400. The population and archive size of 100 models was run for ten cascades of 25 generations each (250 total generations). A single-objective (accuracy) tournament selection was used for both the archive (tourney size of 3) and population (tourney size of 5) model selection. Three types of test problems, small, medium and large were used. The small-sized problem was based on a known analytical function with two inputs and a training set of 100 records generated by random uniform sampling. The medium (8 inputs, 251 datapoints) and large-sized (18 inputs, 1000 datapoints) problems were based on real-life datasets with process noise.

Case I: constant subset and population size - keeping a constant data subset size for all of the cascades with the subset randomly selected for

each generation, actually led to the surprising result of improved Pareto front quality as the data subset size decreased up to 40% of the original data set size. The exception was a problem using data from a designed experiment where each data point was unique and critical.

Case II: increasing subset and constant population size - here the approach was to linearly increase the (random) subset size from 10% to 100% of the data over a number of generations and finish the symbolic regression using the full data set. This produced improved modeling results in comparison to the first case with an interpretation that the smaller subsets in the earlier generations introduced more noise into the modeling process and, therefore, resulted in more exploration - analogous to simulated annealing.

Case III: increasing subset and decreasing population size - in this case the data subset size was linearly increased from 10% to 100% of the data over the first 80% of the generations with the full data set used for the remaining 20% of the generations. The computational effort was kept constant by starting with a model population of 1000 and linearly decreasing it as the data subset size was linearly increased. Effectively, this resulted in a large population for an initial coarse screen shifting to an intensive exploitation with a smaller population in the final generations. This approach was a clear winner both in terms of Pareto front quality and in consistency of modeling results as measured by the standard deviation from 30 independent evolutions. The results for this case are illustrated in Figure 9.

Smits & Vladislavleva also compared **Ordinal ParetoGP** (OPGP) running for 250 generations to conventional **ParetoGP** (PGP) running for 1000 generations for 30 independent evolutions. OPGP produced higher quality and more consistent results than PGP despite only having a quarter of the CPU cycles allocated to it relative to PGP.

Although **ParetoGP** was used in these initial studies, the computational efficiency and accuracy gains should also apply to other flavors of genetic programming. Actually, we should expect greater advantages since those approaches do not have an archive which would also have to be evaluated against the various data subsets for each generation. The previous results on tournament selection and Pareto tournament selection with single and multiple winners offers some of the ingredients to advance the OO theory from Ho *et al* to an 'iterative' OO theory.

6 Conclusions & Summary

In this chapter we have introduced the notion of ordinal optimization and its application to genetic programming. As demonstrated by the significant performance gains of the exploratory investigations, this is a very exciting synergy with much promise for both the practitioner as well as the theoretician since improvements in algorithm efficiency is always welcome to the practitioner and ordinal optimization could provide a new theoretical foundation

Fig. 9. Comparison of ParetoGP with Ordinal ParetoGP - Here we show the results from the increasing subset and decreasing population size case. The initial scatter in quality metric (% area under the Pareto front) for OPGP is due to the differing random subsets used for the first 80% of the generations. Note the improved model ensemble quality (as measured by the area under the Pareto front) relative to the conventional ParetoGP as well as the improved consistency of results

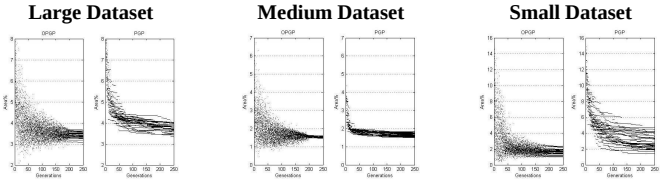


Table 3. Here we compare the Ordinal ParetoGP performance against the conventional ParetoGP for each of the data sets. Note that the OPGP running for 250 generations outperforms the PGP algorithm running for 1,000 generations (40 cascades). This is the basis of the claim of at least a four-fold improvement in algorithm efficiency.

Test Problem	Method	Mean Area%	SD Area%
Small	OPGP 250 gen	1.63	0.32
	PGP 250 gen	2.83	0.90
	PGP 1000 gen	1.69	0.46
Medium	OPGP 250 gen	1.53	0.04
	PGP 250 gen	1.65	0.09
	PGP 1000 gen	1.51	0.08
Large	OPGP 250 gen	3.42	0.14
	PGP 250 gen	3.76	0.19
	PGP 1000 gen	3.49	0.20

for genetic programming as well as guide the development of new algorithms and concepts. The introduction of the **ParetoTourneySelect** method is also significant in that it allows classical GP schemes to be easily migrated to being Pareto-aware. It is also an extension of the single-objective tournament selection method and, therefore, attractive because of the ease of tuning the selection focus and diversity as well as exploiting a local ordinality (the tournament pool) in the selection process - features which are useful as we migrate to an ordinal optimization perspective with an explicit goal of failing forward and initial exploration segueing into a refinement and exploitation stage.

In summary, the future is looking bright for continued advances in the theory, application and impact of GP, in general, and symbolic regression, in particular.

References

1. G. Smits and M. Kotanchek, "Pareto-front exploitation in symbolic regression," in *Genetic Programming Theory and Practice II*, U.-M. O'Reilly, T. Yu, R. L. Riolo, and B. Worzel, Eds. Ann Arbor: Springer, 13-15 May 2004, ch. 17, pp. 283–299.
2. G. Smits, A. Kordon, K. Vladislavleva, E. Jordaan, and M. Kotanchek, "Variable selection in industrial datasets using pareto genetic programming," in *Genetic Programming Theory and Practice III*, ser. Genetic Programming, T. Yu, R. L. Riolo, and B. Worzel, Eds. Ann Arbor: Springer, 12-14 May 2005, vol. 9, ch. 6, pp. 79–92.
3. F. Castillo, A. Kordon, and G. Smits, "Robust pareto front genetic programming parameter selection based on design of experiments and industrial data," in *Genetic Programming Theory and Practice IV*, ser. Genetic and Evolutionary Computation, R. L. Riolo, T. Soule, and B. Worzel, Eds. Ann Arbor: Springer, 11-13 May 2006, vol. 5, ch. 2, pp. –.
4. A. Kordon and C.-T. Lue, "Symbolic regression modeling of blown film process effects," in *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*. Portland, Oregon: IEEE Press, 20-23 June 2004, pp. 561–568.
5. G. Smits and E. Vladislavleva, "Ordinal pareto genetic programming," in *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, vol. 1. IEEE Press, 2006, p. to be published.
6. S. Bleuler, M. Brack, L. Thiele, and E. Zitzler, "Multiobjective genetic programming: Reducing bloat using SPEA2," in *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*. COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea: IEEE Press, 27-30 May 2001, pp. 536–543. [Online]. Available: ftp://ftp.tik.ee.ethz.ch/pub/people/zitzler/BBTZ2001b.ps.gz
7. P. Saetrom and M. Hetland, "Multiobjective evolution of temporal rules." [Online]. Available: citeseer.ist.psu.edu/saetrom03multiobjective.html
8. E. D. de Jong, R. A. Watson, and J. B. Pollack, "Reducing bloat and promoting diversity using multi-objective methods," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, L. Spector, E. D. Goodman, A. Wu, W. B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, and E. Burke, Eds. San Francisco, California, USA: Morgan Kaufmann, 7-11 July 2001, pp. 11–18. [Online]. Available: http://www.demo.cs.brandeis.edu/papers/rbpd_gecco01.pdf
9. J. Hu, E. D. Goodman, and K. Seo, "Continuous hierarchical fair competition model for sustainable innovation in genetic programming," in *Genetic Programming Theory and Practice*, R. L. Riolo and B. Worzel, Eds. Kluwer, 2003, ch. 6, pp. 81–98.
10. Y.-C. Ho, "Soft optimization for hard problems" computerized lecture via private communication/distribution." [Online]. Available: http://www.hrl.harvard.edu/~ho/DEDS/OO/OOTOC.html

Index