Hybridisation of SMS-EMOA with Regression Models

740036129

ABSTRACT

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

key-terms: HVC Hypervolume Contribution, **SMS-EMOA**

1 INTRODUCTION

A key distinction to make is between Hypervolume and HVC these are linked but HVC builds upon the concept of hypervolume and while using it allows us to distinguish between solutions/individuals that do not contribute much to the whole Hypervolume as a whole.

When talking about hypervolume when we really mean area for the *ZDT* sets as they use a 2 dimensional objective spaces. TODO: FINISH THIS LINE

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

introduction text goes here

This whole paper is written with the assumption that hypervolume calculations are computationally expensive.

2 NUMERICAL EXPERIMENTS

This sections is divided into two major subsections. The first discussing the implemented hybridisation of SMS-EMOA 2.1, The second covering default behaviour of SMS-EMOA and the comparison metrics that are utilised and discussed throughout this paper.

2.1 Hybridisation of SMS-EMOA

In SMS-EMOA works based on the *hypervolume* indicator. However it does not actually calculate the actual hypervolume as a whole, instead it calculates the HVC as a part of choosing what parts of the population survives in any given generation. As such the survival function is parsed fronts and returns a number representing an individual's contribution to the population as a whole, this is what the regression model will be used to predict.

As such the regression model(s) do not directly interact with the decision variables instead models the contribution of a given individual on the objective space. It would be great if we could parse in a model that replaces HVC that knows the objective landscape before we start optimising, and yes with the *test problems* this can be done but in real situations this is *not always known* as such a hybrid approach is needed where a proportion of the time it is working based on actual HVC calculations and the remaining time it is working based upon a model predicting these HVCs. Here we found that roughly a 50:50 split flipping between prediction and actual calculations every 10 fronts.

It was found that working with a flip rate of less than 10 fronts would mean that the model impede convergence towards a diverse set of solutions, this would often cause large gaps or cover only one section of the Pareto front in all the test sets.

Algorithm 1 Hybrid Survival Function Pseudo Code

```
if bool_eval_type == False then
    # fit the model against real contribution
    hypervolume_contribution = Calcualte_Actual(front)
    fit_regression_model(front, hypervolume_contribution)
else bool_eval_type == True
    # predict using the model (Aprox half the time)
    hypervolume_contribution = predict(front)
end if

if eval_counter >= 10 then
    eval_conter = 0 # rest counter
    bool_eval_type = (not bool_eval_type) # flip
else
    eval_conter++ # increment counter
end if... (default behaviour)
```

2.2 SMS-EMOA Default Behaviour and Metrics

The number of Decision Variables tested for each set are 5, 10 or 30. Test Sets: *ZDT1*, *ZDT3*, *ZDT4*, *ZDT6*

Test Set	Number of Decision Variables	Hypervolume	Generations to Convergence
ZDT1	5	0.8706010094731254	40
	10	0.8701004215097737	70
	30	0.8667542151989153	140
ZDT3	5	1.326932799989349	40
	10	1.3251498767866068	70
	30	1.3132467347404106	120
ZDT4	5	0.8711174340219447	130
	10	0.8630376835162556	190
	30	0.8508551733759164	670
ZDT6	5	0.4959617673719027	130
	10	0.4903102556919033	230
	30	0.4898010767283024	600

Note: all of the hypervolume/areas calculated within this paper are using the reference point of $(1.1, 1.1)$. The generations to convergence were calculated to nearest 10 and are selected via visual inspecting of the front against the real test set Pareto front.

These values in the table above is the primary data used for comparisons and evaluating the models used in the remainder of this report as they are used as what the model should produce given the same input, obviously with predictions that not possible but these are the best case scenarios.

2.2.1 The Similarity Metric. Represents how well the hybrid version of SMS-EMOA performed against the original model.

$$similarity = 1 - \frac{original\ SMSMOEA\ hypervolume}{total\ potential\ calls}$$

$0 < similarity \leq 0.90$ - poor representation with large gaps in the front or only cover a small portion of the front.

$0.9 < similarity \leq 0.995$ - decent results here most of the front is covered with but there is very poor diversity and potentially some quite large gaps

$0.995 < similarity \leq 0.99$ - no major gaps in the solution but either has a small gap at one side of the front is not distributed evenly. Overall these are a very good set of solutions.

$0.99 < similarity \leq 1.01$ - should be considered on-par with the original SMS-EMOA for this test.
Values greater than 1.00 can be found are perform better than the original model, these are due to a quirk in how the specific models work and the shape of the Pareto front - just look at the standard Linear Regression for *ZDT6* with 5 decision variables it achieves a similarity score of 1.0037 (Table 1).

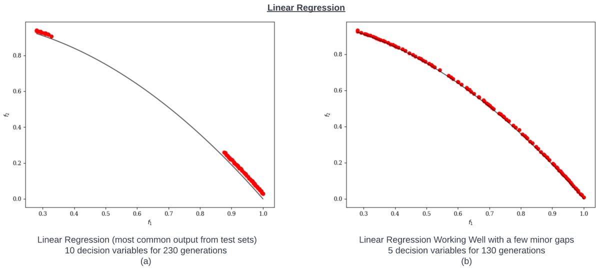
2.2.2 Call Reduction Metric. represents the percentage of how many HVC calls for an entire front are reduced when comparing against how many calls would have been made if the model was not there (this is an estimate based on a counter that increments when any front is calculated or estimated).

$$call\ reduction = \frac{acutal\ calls}{total\ potential\ calls}$$

The results discussed can be seen in Table 1, more than this were generated but these are a selection of the best found models and some interesting observations (this is discussed in the next section).

3 ANALYSIS

3.1 Linear Regression: an interesting observation



Linear Regression (most common output from test sets)
10 decision variables for 230 generations
(a)

Linear Regression Working Well with a few minor gaps
5 decision variables for 130 generations
(b)

Figure 1: ZDT6 (Table 1) Linear Regression for 10 and 5 decision variables

In most of the tests, as seen in figure 1a this was the general result, often it would get stuck to one side of the Pareto front resulting in a coverage of only a small segment. This likely occurs due to it reinforcing due to the regression technique fitting a linear line between all found points, all the test sets are have non-linear Pareto fronts. As such, what ends up happening is that it either fits itself to one extreme acting similar to a tangent or two where most of it is in front or behind the Pareto front what then pushes solutions to extremes either end.

During testing it was found that once it converges upon an extreme, the EA pushes itself slows towards more extreme sets of solutions with a progressive lowering of diversity. This in part is due to how this the EA works in the *pymoo* library with no active recall to the best previous solution (as it does not calculate overall hypervolume instead works on a selection and rejection process for the population based upon the hybrid HVC and model predictions.

Linear regression fits well to *ZDT6* as shown in figure 1b and for 30 decision variables overall it performs decently compared to how it usually performs. This likely occurs on *ZDT6* especially as there is such a small change in the gradient as such a linear line could be fit close to it.

3.2 Random Forrest

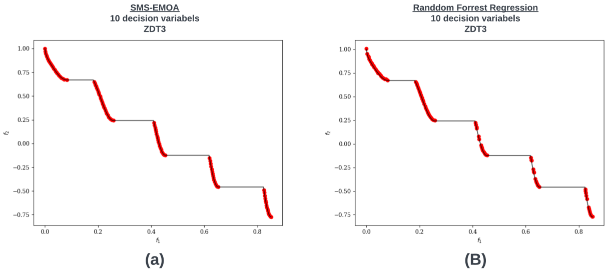


Figure 2: Random Forrest in one of the few cases it performs well

Figure 2 shows the RF, it performs really well on highest two steps and performs worst on the 3rd and 4th one down and only has a minor gap in the middle of the final lowest step.

Overall RF performs very well on all sets for 10 decision variables and much worse on 5 and 30.

3.3 Support Vector Regression (SVR): the best for modelling hypervolume (found*)

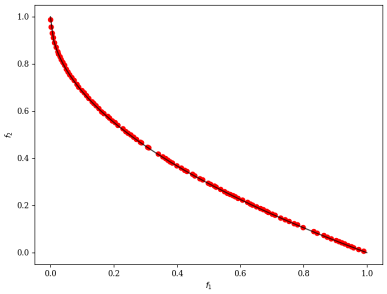


Figure 3: Illustrates Support Vector Regression with the poly kernal *sklearn*

Support Vector Regression (SVR) works incredibly well for all test sets excluding *ZDT4*. In these three of these cases all performing better than Random Forrest. Throughout all of the solutions produced using SVR (kernal Poly or RBF) are generally well distributed along the fronts (or section of the front in the case *ZDT4*).

In figure 3 it can be seen that it generally finds the front and has a good distribution across it with only two small points where there is not the greatest diversity, however comparing this is using 44.12% of the original calls for calculating HVC, this should be considered very good and even matches 0.9964 (pretty much within 0.36% similarity to the overall hypervolume of the original SMS-EMOA model for the same number of generations).

Why is SVR working so well to model and learn the space based on previous hypervolume contribution calculations? Well this is likely because SVR creates support vectors in the space and can then use them to guide the regression line across these points. Using the kernels RFB or Poly (degree of 3 here is set so up to (so a cubic relationship is drawn)) allow for these points to map a non-linear regression line.

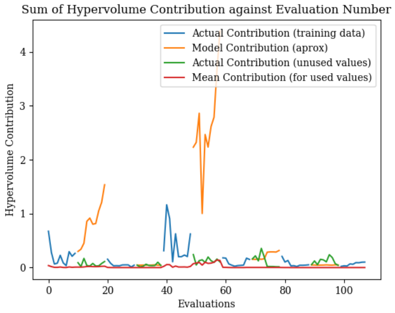


Figure 4: A continuation of the previous figure 3 showing the Sum of the Hypervolume Contribution (for any given set of Evaluations)

In the figure above this highlights some interesting, this shows that the zdt1 for 10 decision variables (Table 1 and fig 3 / 4) has some issues with modelling this data at certain points, ultimately this could potentially introduce some error but at these points the model is still improving this therefore suggests that where it continues on from the training set (seen in blue) the model is over predicting massively - these over predictions as they are only used as selection criteria does not seem to have a major effect on the results.

The first model prediction set (orange) occurs most of the time, and this is likely due to the dataset chaining so much during its set of 10 predictions as such it starts over predicting when its converging. The remainder of the set where it is not over predicting, other than it progressively gets closer to the mean contribution, suggesting that as it begins to settle along Pareto front the distribution is getting better.

4 CONCLUSION

In conclusion SVR using either the Poly or the RBF kernal perform best for mimicking the Pareto front found through training on hypervolume contribution data while the EA is exploring the area. If you have 10 decision variables RF appears to be the way to go as it got pretty good at modelling the contribution for those.

Overall the reduction is generally very good and tends towards 50% (this is likely because the *pymoo* implantation does not call the survival function every generation this can only be assumed to be due to a generation being too small?) but for the tests often fall more towards 45%

REFERENCES

Table 1: Performance Against unaltered SMS-EMOA *pymoo* & *sklearn*

		Decision		Final (4.s.f)	Actual	Potential	Similarity (1)	call
Regression Model	Test Set	vars	Generations	Hypervolume	eval calls	eval calls	(4.s.f)	reduction (4.s.f)
Random Forest	ZDT1	5	40	0.7822	20	37	0.8985	45.95%
		10	70	0.8667	38	68	0.9961	44.12%
		30	140	0.8463	70	136	0.9764	48.53%
	ZDT3	5	40	1.2326	20	37	0.9289	45.95%
		10	70	1.3228	39	69	0.9982	43.48%
		30	120	1.2948	60	116	0.986	48.28%
	ZDT4	5	130	0.773	65	125	0.8874	48.00%
		10	190	0.8049	91	181	0.9326	49.72%
		30	670	0.829	309	609	0.9743	49.26%
	ZDT6	5	130	0.4908	64	124	0.9896	48.39%
		10	230	0.4854	114	224	0.99	49.11%
		30	600	0.4721	300	595	0.9638	49.58%
SVR Kernal = Poly	ZDT1	5	40	0.8653	20	39	0.9939	48.72%
		10	70	0.869	39	69	0.9987	43.48%
		30	140	0.8609	70	137	0.9933	48.91%
	ZDT3	5	40	1.215	20	36	0.9156	44.44%
		10	70	1.3218	39	69	0.9975	43.48%
		30	120	1.3075	60	116	0.9956	48.28%
	ZDT4	5	130	0.8203	67	127	0.9416	47.24%
		10	190	0.8054	91	181	0.9332	49.72%
		30	670	0.8196	320	632	0.9633	49.37%
	ZDT6	5	130	0.4955	67	127	0.999	47.24%
		10	230	0.4869	112	222	0.993	49.55%
		30	600	0.4877	293	583	0.9957	49.74%
SVR Kernal = RBF	ZDT1	5	40	0.8649	20	39	0.9935	48.72%
		10	70	0.8684	39	69	0.9981	43.48%
		30	140	0.8609	70	137	0.9933	48.91%
	ZDT3	5	40	1.2364	20	35	0.9318	42.86%
		10	70	1.3238	39	69	0.999*	43.48%
		30	120	1.3109	60	119	0.9982	49.58%
	ZDT4	5	130	0.8203	67	127	0.9416	47.24%
		10	190	0.82	91	181	0.9501	49.72%
		30	670	0.8411	320	631	0.9886	49.29%
	ZDT6	5	130	0.495	66	126	0.998	47.62%
		10	230	0.4806	110	216	0.9803	49.07%
		30	600	0.4834	294	584	0.9869	49.66%
Linear	ZDT6	5	130	0.4978	67	127	1.0037	47.24%
		10	230	0.3385	110	220	0.6903	50.0%
		30	600	0.4832	298	588	0.9866	49.32%