MSPOT: Multi Criteria Optimization with the Sequential Parameter Optimization Toolbox (SPOT Version 1.0.2166)

T. Bartz-Beielstein, B. Naujoks, M. Zaefferer

March 8, 2012

1 Introduction

Up to version 0.1.1550, SPOT [1] was designed and could only be applied to single objective optimization problems. Despite this, many real world applications feature more than just one quality criterion. Therefore, SPOT was extended to be applicable to Multi Criteria Optimization (MCO). This application of SPOT is referred to as MSPOT.

The basic principle of SPOT remains the same for MCO problems. An initial design is created based on sampling in the decision space, e.g, by a Latin Hypercube Design (LHD) design. Based on the information provided by the target functions (i.e. more than just one objective), several surrogate models are build, one for each objective of the MCO problem. The models are exploited to suggest new design points for evaluation, which then yield new information from the target function. Based on this new information, a new model can be build. This process is iterated until stopped by some termination criterion.

Several surrogate models in SPOT can be used for MCO. To exploit the generated models, two tools are available in MSPOT. The first one is the naive sampling approach, the second one is the utilization of typical MCO algorithms. Both will be demonstrated in this document.

The following section introduces the MCO test function utilized in this document and presents some optimization performed with alternative MCO algorithms.

2 The ZDT2 Testfunction

The test function used in the following examples is part of the "mco" R-package[1]. In this example, it is expected that this package is already installed and loaded. The mco package also contains the NSGA-II [2] algorithm, which

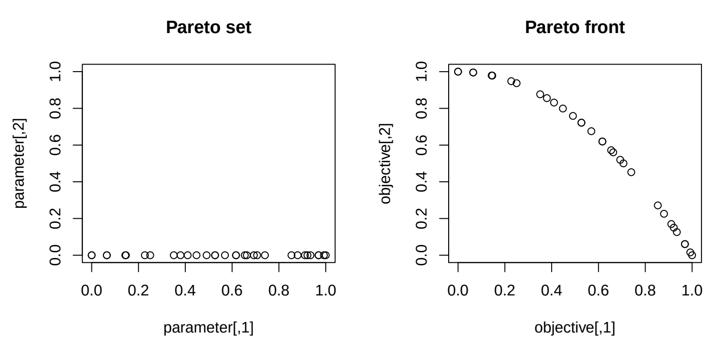
[1] http://cran.r-project.org/web/packages/mco/

is used as an alternative optimizer. However, it can optionally be employed in MSPOT as well. SPOT is loaded since it will be used in turn, of course.

```
> #install.packages("mco")
> #install.packages("SPOT")
> require("mco")
> require(SPOT)
```

The ZDT functions [3] are defined on a normalized decision space, i.e. $[0;1]^n$. The dimension n can be freely chosen. To guarantee a low number of function evaluations as well as easy visualization of results, both, the dimension of the decision and the objective space are set to 2 in following example. To show the behavior of the ZDT2 test function, it is firstly optimized with the NSGA-II algorithm.

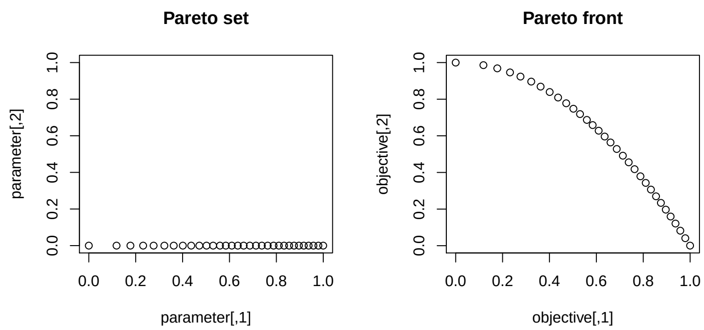
```
> resNSGA<-nsga2(zdt2,2,2,lower.bounds=c(0,0),
+          upper.bounds=c(1,1),popsize=32,generations=100)
> par(mfrow=c(1,2))
> objective=resNSGA$value
> parameter=resNSGA$par
> plot(parameter, main="Pareto set",ylim=c(0,1))
> plot(objective, main="Pareto front")
```



This indicates how the Pareto front approximately looks like. The Pareto set depicts, where the Pareto optimal solutions are located in the decision space. Here, they reside on the lower boundary of the second parameter. Of course, any other adequate algorithm can be used for optimization as well. The following code calls a straight-forward SMS-EMOA [4], which is shipped with the SPOT package.

```
> resSMS <- spotSmsEmoa(zdt2,
+          lower=c(0,0),
+          upper=c(1,1),
+          control=list(mu=32,maxeval=3200))
> par(mfrow=c(1,2))
```

```
> objective=t(resSMS$value)
> parameter=t(resSMS$par)
> plot(parameter, main="Pareto set",ylim=c(0,1))
> plot(objective, main="Pareto front")
```



Both, SMS-EMOA and NSGA-II approximate the Pareto front rather well with 3 200 function evaluations. The distribution of points on the Pareto front looks much more regular for the SMS-EMOA. This indicates that the neighbouring points in the Pareto set are getting closer for higher values of the first parameter.

Using a reference point, the hypervolume can be computed for both Pareto fronts received from SMS-EMOA and NSGA-II.

```
> volNSGA<-dominated_hypervolume(t(resNSGA$value),c(2,2))
> volNSGA
```

```
[1] 3.307559
```

```
> volSMS<-dominated_hypervolume(resSMS$value,c(2,2))
> volSMS
```

```
[1] 3.318701
```

For this single experiment, the Pareto front of the NSGA-II algorithm has a larger hypervolume. In the following chapter, the question how MSPOT does compare to these is examined.

3 The naive sampling approach

One way to do multi objective optimization with SPOT, is to exploit the surrogate models by evaluating a large LHD on them. The "best" points of the design will be suggested for evaluation on the target function. In this context, "best" is defined to be the lowest dominated sorting rank. If the rank of several points is the same, the hypervolume contribution of each single point will be considered to choose between them.

To test this approach with MSPOT, a configuration list is created first:

```
> config=list()
```

The above algorithms uses 3 200 function evaluations. This is quite a lot for SPOT, as building the models is rather expensive. In fact SPOT is mostly used in problems that use only a small number of function evaluations, like industrial real world applications.

Therefore, the budget for SPOT is restricted to just 100 evaluations:

```
> config$auto.loop.nevals=100
```

Next, the size of the large LHD is specified, we consider 1 000 design points here.

```
> config$seq.design.size=1000
```

In each sequential SPOT step, a certain number of design points will be evaluated. In this case, 10 points are chosen in each step, leading to 32 steps over all.

```
> config$seq.design.new.size=10
```

Since the invoked test function is not noisy, old design points do not have to be reevaluated. As a consequence, repeats in the sequential or initial design are not needed. Furthermore, SPOT's OCBA [5] feature should not be used.

```
> config$seq.design.oldBest.size=0
> config$spot.ocba=FALSE
> config$seq.design.maxRepeats = 1
> config$init.design.repeats = 1
```

Two functions have to be chosen in the list. The first function is the surrogate model interface. For multi objective optimization "spotPredictForrester", "spotPredictMlegp", "spotPredictEarth", "spotPredictRandomForest" and "spotPredictLm" are available. Since it is fast and robust, the Multivariate Adaptive Regression Spline [6] Model ("spotPredictEarth") is selected.

The second function specifies how the surrogate model is optimized. This is NA in this case, because only the sampling approach is used. Alternatively, it can be "spotParetoOptMulti", which will be demonstrated later in this document.

```
> config$seq.predictionModel.func="spotPredictEarth"
> config$seq.predictionOpt.func<-NA
```

Finally, SPOT needs some information about the target function. Its region of interest, in which the parameters are varied, has to be specified, as well as the name of target function itself.

```
> config$alg.func=zdt2
> config$alg.roi=spotROI(lower=c(0,0),upper=c(1,1))
```

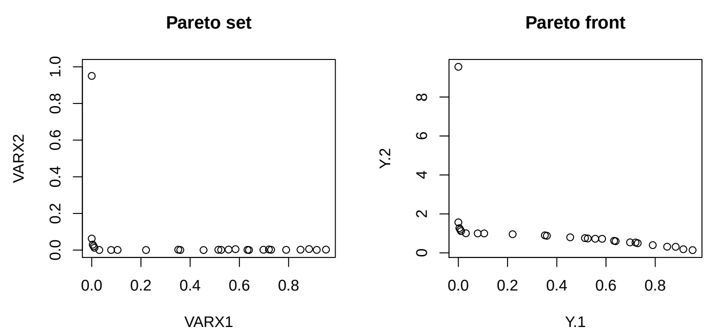
Now, using the above configuration, SPOT can be started.

```
> res1<-spot(spotConfig=config)
```

```
spot.R::spot started
```

The results can be evaluated as follows.

```
> par(mfrow=c(1,2))
> objective=res1$mco.val
> parameter=res1$mco.par
> plot(parameter, main="Pareto set",ylim=c(0,1))
> plot(objective, main="Pareto front")
```



4 Optimization of the surrogate models

The results observed in the previous section were far from good. Although they were based on a low number of function evaluations, they can be improved by choosing better settings. Therefore, SMS-EMOA is chosen to optimize the surrogate models. Moreover, instead of creating a large design, SMS-EMOA is provided with a large budget.

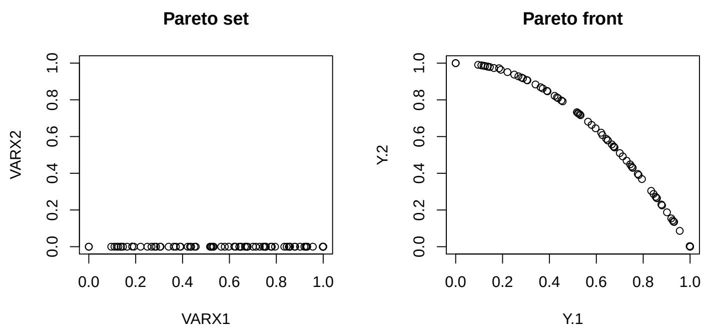
```
> config$seq.design.size=10
> config$seq.predictionOpt.func="spotParetoOptMulti"
> config$seq.predictionOpt.method="sms-emoa"
> config$seq.predictionOpt.budget=1000
> config$seq.predictionOpt.psize=20
```

SPOT is started again with the altered configuration.

```
> res2<-spot(spotConfig=config)
```

```
spot.R::spot started
```

```
> par(mfrow=c(1,2))
> objective=res2$mco.val
> parameter=res2$mco.par
> plot(parameter, main="Pareto set",ylim=c(0,1))
> plot(objective, main="Pareto front")
```




```
> volsamp<-dominated_hypervolume(t(res1$mco.val),c(2,2))
> volopt<-dominated_hypervolume(t(res2$mco.val),c(2,2))
> volNSGA

[1] 3.307559

> volSMS

[1] 3.318701

> volsamp

[1] 3.138789

> volopt

[1] 3.321052
```

As can be seen from the example above, the hypervolume of the elaborate approach is even slightly better than the one of the straight-forward SMS-EMOA. Of course, this is only a single experiment and would have to be reevaluated several times with different seeds to gain statistical significance. Still, it has to be noticed that MSPOT uses 100 evaluations of the target function only. Moreover, a complete archive of all non dominated solutions is kept for MSPOT. For a fair comparison, this should be compared against a similar archive of SMS-EMOA and NSGA-II, instead of comparing it against the final populations.

5 Outlook: Future Development

As can be seen from the last MSPOT run, the distribution of points on the Pareto front is not as nice and regular as the one received from the straightforward SMS-EMOA. This is one thing where MSPOT can be improved. At the moment, MSPOT simply considers the results from sampling and the optimization on the surrogate, when selecting new points for being evaluated on the target function. Ideally, MSPOT should also consider already known points (i.e., points that were already evaluated on the target function).

Furthermore, when using optimization algorithms on the surrogate models, the points sampled with the LHD are only used to supplement the design points found by the surrogate optimization, if there are not enough. It might be profitable to use the LHD results to produce start populations for the internal optimization on the surrogate models.

Another interesting application for MSPOT can be many objective optimization problems. The information gained from the surrogate models could be employed to reduce the number of objectives that need to be considered.

References

[1] Bartz-Beielstein, T.; Lasarczyk, C., and Preuss. Sequential Parameter Optimization. In *Proceedings 2005 Congress on Evolutionary Computation (CEC'05)*, pages 773-780, 2005.

[2] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms.* Wiley, New York, 2001.

[3] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, 2000.

[4] N. Beume, B. Naujoks, and M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.

[5] C. H. Chen. An effective approach to smartly allocate computing budget for discrete event simulation. In *Proceedings of the 34th IEEE Conference on Decision and Control*, pages 2598–2605, 1995.

[6] J. H. Friedman. Multivariate adaptive regression splines. *Ann. Stat.*, 19(1):1–141, 1991.