Analysing Autotuning Results on the SPAPT Benchmark

Pedro Bruel

March 22, 2019

We started analyzing the results obtained applying an automated version of our approach in the SPAPT benchmark, which were presented in our CCGRID paper. The objective of this analysis was to identify automated decisions that would not have been made, or that would have been made differently, if user input was available.

This is an ongoing work, and figures and text are in a preliminary state. This document is organized as follows. Section 1 presents each eliminated factor, and their set values, over the 10 autotuning runs of the `bicg` SPAPT kernel. Section 2 presents a detailed look at the decisions taken during each of the 4 steps performed during the autotuning run, and decisions are discussed based on the ANOVA tables presented.

1 Eliminated Factors and Fixed Values

Figure 1 shows the eliminated model terms and program factors at each of the 4 steps of the 10 autotuning runs. Figure 1 also shows the levels chosen for each factor over all 10 runs. The two top plots on Figure 1 show eliminated model terms and design factors at each step. The third plot shows what were the values fixed after each of the 10 repetitions of the 4 steps. Each factor uses the same color scale to represent its levels, which is reset for each factor.

The binary factors that were always identified as significant, `OMP` and `SCR`, had the same fixed value in all repetitions. The next three most eliminated factors were `RT_I`, `T2_I`, and `T1_I`, which all had linear, quadratic, and cubic terms in the initial model.

The level selected for the `RT_I` factor was `RT_I=8` 5 times, followed by `RT_I=4` 3 times, and `RT_I=1` and `RT_I=2` one time each. This factor has 6 levels, which are powers of 2 from 1 to 32, and the fixed values favored the middle of the scale. It could be worth it to see if it would be possible to add more levels to this factor, specially if non-powers of two are allowed.

We also see in the topmost figure that the model term for `RT_I` most frequently responsible for eliminating the factor was the linear term, while the quadratic term was responsible for the elimination 2 times, and the cubic term was never responsible. Further experimentation with this kernel could remove the cubic term.

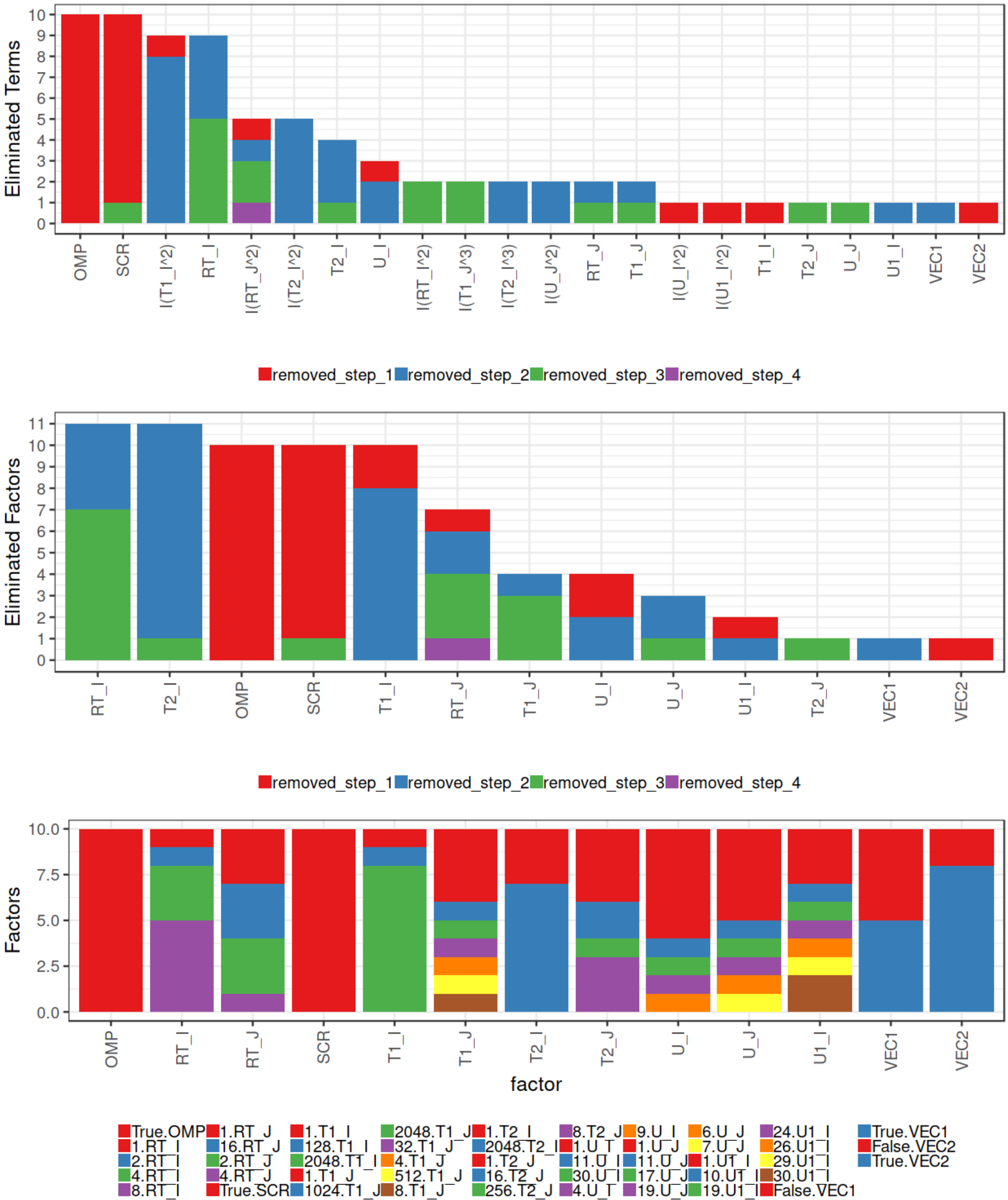

Figure 1: Different visualizations of eliminated factors and set levels

2 Choices at Each Step

In this section we will look at the choices made by the automated version of DLMT at each of the 4 steps of one of the 10 optimization iterations performed for `bicgkernel`.

The initial model contained linear, quadratic and cubic terms for all applicable factors of `bicgkernel`. The Pr(> F) threshold was set to 0.05, or "one star".

With hindsight, obtained from knowing the most eliminated variables from Section 1, we can argue that a missing fundamental user-centered ability of our approach is that of being capable of going back to the model of a previous iteration, that is, reverting a factor elimination if we reach a point were nothing seems to be significant.

Reinforcing the argument for being able to revert iterations, a possible error identified from looking at these logs is that the linear terms are not added to the prediction step models when a quadratic term is identified as significant, but the linear term is not. The same happens for quadratic and linear terms not identified as significant when a cubic term is significant. On one hand, it seems that it is correct to not use terms not identified as significant on the prediction model, but on the other hand, it seems that by doing so we are trusting too much on the generality of the sample for which we ran the ANOVA test, plus, we might lose the flexibility added by the other terms with the same factor.

Another useful capability would be adding extra experiments to an existing design in order to explore more levels of ambiguous factors. Choosing to add more experiments could provide more information at the cost of more measurements, but could avoid loosing an undetected significant factor or choosing misidentified ones.

The next 4 sections present the ANOVA tables for each step. Sections 2.1, 2.2 and 2.3 are annotated with preliminary analyses of the automated decisions. A complete version of this document will provide annotations and more detailed discussions of each decision. The final objectives are having a list of useful user-centered features and a demonstration of how user input can improve the results.

2.1 First Step

The source blocks in the following sections show excerpts of the log file for one of the 10 repetitions of autotuning the `bicgkernel` SPAPT kernel. The excerpts are annotated with line numbers for easy referencing.

This is the ANOVA table for the first design generated from a set of uniformly generated points:

```
Regression Step: Df Sum Sq Mean Sq F value  Pr(>F)
I(T1_I^2)     1   4.47    4.47   2.624 0.15642
I(T1_J^2)     1   3.00    3.00   1.762 0.23262
I(T2_I^2)     1   9.32    9.32   5.471 0.05792 .
I(T2_J^2)     1   0.00    0.00   0.001 0.98132
I(U1_I^2)     1   5.23    5.23   3.069 0.13037
I(U_I^2)      1   0.38    0.38   0.223 0.65324
I(U_J^2)      1   4.92    4.92   2.887 0.14020
I(RT_I^2)     1   0.16    0.16   0.091 0.77302
I(RT_J^2)     1  17.78   17.78  10.433 0.01791 *
T1_I          1   2.65    2.65   1.554 0.25901
T1_J          1   9.66    9.66   5.669 0.05469 .
T2_I          1   4.55    4.55   2.671 0.15331
T2_J          1   0.06    0.06   0.036 0.85481
U1_I          1   8.90    8.90   5.224 0.06231 .
U_I           1   2.58    2.58   1.512 0.26483
U_J           1   1.26    1.26   0.741 0.42233
RT_I          1   5.55    5.55   3.256 0.12121
RT_J          1   0.63    0.63   0.371 0.56470
SCR           1  16.77   16.77   9.845 0.02013 *
VEC1          1   0.24    0.24   0.143 0.71808
VEC2          1   2.22    2.22   1.304 0.29700
OMP           1  57.31   57.31  33.636 0.00115 **
I(T1_I^3)     1   0.00    0.00   0.000 0.99334
I(T1_J^3)     1   0.88    0.88   0.518 0.49865
I(T2_I^3)     1   1.47    1.47   0.861 0.38916
I(T2_J^3)     1   2.30    2.30   1.347 0.28983
I(U1_I^3)     1   0.04    0.04   0.023 0.88542
I(U_I^3)      1   0.04    0.04   0.025 0.87986
I(U_J^3)      1   0.01    0.01   0.004 0.95158
I(RT_I^3)     1   1.76    1.76   1.031 0.34910
I(RT_J^3)     1   0.20    0.20   0.116 0.74534
Residuals     6  10.22    1.70
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The 3 factors identified here, `RT_J`, `SCR`, and `OMP` are in the top 6 most eliminated factors for this kernel. This is the only one of the 10 repetitions where `RT_J` was eliminated in the first step. We proceed to refit the model using only the significant factors to the same design data used in the first linear regression. We then generate a new uniform sample to predict the best point using this new fitted model. An idea to discuss is whether it would be useful to generate a new design for the significant factors only, and then fitting the pruned new model using this new design. This could enable us to benefit more from cases where the significance was identified correctly.

```
Using Model: . ~ OMP + I(RT_J^2) + SCR
Prediction Regression Step:            Df Sum Sq Mean Sq F value   Pr(>F)
OMP            1  79.57   79.57   59.63 5.57e-09 ***
I(RT_J^2)      1  20.59   20.59   15.43 0.000398 ***
SCR            1  29.03   29.03   21.75 4.67e-05 ***
Residuals     34  45.37    1.33
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Next, we see that our best prediction was worse than the best point measured in our design. This does is not really damaging to our case, since we are not deeply interested in the model's accuracy, but simply in exploring better regions of the search space. The budget ratio is not relevant for this experiment, but it keeps track of how many experiments were actually evaluated.

```
Slowdown (Design Best): 0.097612475844
Slowdown (Predicted Best): 0.149470859652
Budget: 38/1000
Best point from design was better than predicted point
```

The next excerpt shows the levels selected for each factor. We see that `RT_J` was set to its first level `RT_J=1`, indexed by the 0th element of a Python array. This is one of the 3 most common levels for `RT_J` in this kernel, which are 1, 2 and 16. Factors `OMP` and `SCR` were set to the only level observed in this experiment, which is `True`. It was therefore always easy to identify the significance of these factors, as well as their impact on performance. This could be an argument for going back and not fixing the `RT_J` factor, since its significance was smaller than the other 2 factors. With the 2 most significant factors out of the way, the significance of other factors could be measured better, and their levels predicted better.

```
Current Model: {'inverse': [], 'linear': ['T1_I', 'T1_J', 'T2_I',
'T2_J', 'U1_I', 'U_I', 'U_J', 'RT_I', 'VEC1', 'VEC2'], 'cubic': ['T1_I',
'T1_J', 'T2_I', 'T2_J', 'U1_I', 'U_I', 'U_J', 'RT_I'], 'interactions': [],
'fixed_factors': {'OMP': 1, 'RT_J': 0, 'SCR': 1}, 'quadratic': ['T1_I', 'T1_J',
'T2_I', 'T2_J', 'U1_I', 'U_I', 'U_J', 'RT_I'], 'response': 'cost_mean'}
```

2.2 Second Step

```
Regression Step:            Df  Sum Sq Mean Sq F value  Pr(>F)
I(T1_I^2)     1 0.11484 0.11484  42.840 0.00725 **
I(T1_J^2)     1 0.00433 0.00433   1.615 0.29334
I(T2_I^2)     1 0.04000 0.04000  14.923 0.03067 *
I(T2_J^2)     1 0.00594 0.00594   2.214 0.23346
I(U1_I^2)     1 0.00130 0.00130   0.485 0.53613
I(U_I^2)      1 0.01305 0.01305   4.869 0.11446
I(U_J^2)      1 0.00739 0.00739   2.757 0.19542
I(RT_I^2)     1 0.00567 0.00567   2.116 0.24180
T1_I          1 0.00066 0.00066   0.246 0.65394
T1_J          1 0.00049 0.00049   0.184 0.69720
T2_I          1 0.01692 0.01692   6.313 0.08674 .
T2_J          1 0.00324 0.00324   1.210 0.35168
U1_I          1 0.00018 0.00018   0.067 0.81317
U_I           1 0.00108 0.00108   0.403 0.57071
U_J           1 0.00007 0.00007   0.027 0.87924
RT_I          1 0.02256 0.02256   8.414 0.06247 .
VEC1          1 0.00037 0.00037   0.138 0.73520
VEC2          1 0.00050 0.00050   0.188 0.69422
I(T1_I^3)     1 0.00292 0.00292   1.088 0.37360
I(T1_J^3)     1 0.00066 0.00066   0.245 0.65440
I(T2_I^3)     1 0.00688 0.00688   2.565 0.20756
I(T2_J^3)     1 0.00040 0.00040   0.148 0.72611
```

```
2202 I(U1_I^3)    1 0.02363 0.02363   8.816 0.05910 .
2203 I(U_I^3)     1 0.00030 0.00030   0.113 0.75893
2204 I(U_J^3)     1 0.00023 0.00023   0.084 0.79033
2205 I(RT_I^3)    1 0.00374 0.00374   1.395 0.32262
2206 Residuals    3 0.00804 0.00268
2207 ---
2208 Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
2257 Using Model: . ~ I(T1_I^2) + I(T2_I^2)
2258 Prediction Regression Step:            Df  Sum Sq Mean Sq F value   Pr(>F)
2259 I(T1_I^2)    1 0.11484 0.11484  22.500 6.07e-05 ***
2260 I(T2_I^2)    1 0.03275 0.03275   6.416   0.0174 *
2261 Residuals   27 0.13781 0.00510
2262 ---
2263 Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

2.3 Third Step

```
3024 Regression Step:            Df  Sum Sq Mean Sq F value  Pr(>F)
3025 I(T1_J^2)    1 0.00811 0.00811   3.970 0.11710
3026 I(T2_J^2)    1 0.01098 0.01098   5.379 0.08121 .
3027 I(U1_I^2)    1 0.00206 0.00206   1.009 0.37191
3028 I(U_I^2)     1 0.00077 0.00077   0.375 0.57346
3029 I(U_J^2)     1 0.00137 0.00137   0.669 0.45923
3030 I(RT_I^2)    1 0.03454 0.03454  16.918 0.01469 *
3031 T1_J         1 0.00000 0.00000   0.000 0.99613
3032 T2_J         1 0.00001 0.00001   0.004 0.94986
3033 U1_I         1 0.00001 0.00001   0.005 0.94958
3034 U_I          1 0.00283 0.00283   1.385 0.30457
3035 U_J          1 0.00058 0.00058   0.286 0.62124
3036 RT_I         1 0.05209 0.05209  25.512 0.00723 **
3037 VEC1         1 0.00001 0.00001   0.003 0.96096
3038 VEC2         1 0.00012 0.00012   0.057 0.82266
3039 I(T1_J^3)    1 0.00103 0.00103   0.505 0.51654
3040 I(T2_J^3)    1 0.00422 0.00422   2.069 0.22368
3041 I(U1_I^3)    1 0.00014 0.00014   0.067 0.80893
3042 I(U_I^3)     1 0.00000 0.00000   0.000 0.98505
3043 I(U_J^3)     1 0.00009 0.00009   0.043 0.84668
3044 I(RT_I^3)    1 0.00728 0.00728   3.564 0.13210
3045 Residuals    4 0.00817 0.00204
3046 ---
3047 Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
3091 Using Model: . ~ RT_I + I(RT_I^2)
3092 Prediction Regression Step:            Df  Sum Sq Mean Sq F value   Pr(>F)
3093 RT_I         1 0.01600 0.01600   7.911   0.0101 *
3094 I(RT_I^2)    1 0.07390 0.07390  36.550 4.38e-06 ***
3095 Residuals   22 0.04448 0.00202
3096 ---
3097 Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
3181 Slowdown (Design Best): 0.0938185300331
3182 Slowdown (Predicted Best): 0.100979130227
3183 Budget: 93/1000
3184 Best point from design was better than predicted point
```

The factor `RT_I` the most identified as significant, considering all its model terms. It was mostly eliminated at the third step, which is the case here. Despite that, the level selected here was the only the second most chosen level `RT_I=4`, the third element in a Python array, indexed by 2.

```
  3227 Current Model: {’inverse’: [], ’linear’: [’T1_J’, ’T2_J’, ’U1_I’,
’U_I’, ’U_J’, ’VEC1’, ’VEC2’], ’cubic’: [’T1_J’, ’T2_J’, ’U1_I’, ’U_I’, ’U_J’],
```

'interactions': [], 'fixed_factors': {'T1_I': 11, 'RT_I': 2, 'T2_I': 11, 'OMP': 1, 'SCR': 1, 'RT_J': 0}, 'quadratic': ['T1_J', 'T2_J', 'U1_I', 'U_I', 'U_J'], 'response': 'cost_mean'}

2.4 Fourth Step

```
Regression Step:            Df   Sum Sq  Mean Sq F value Pr(>F)
I(T1_J^2)    1 0.000005 0.000005   0.001  0.982
I(T2_J^2)    1 0.001378 0.001378   0.176  0.703
I(U1_I^2)    1 0.000101 0.000101   0.013  0.917
I(U_I^2)     1 0.000493 0.000493   0.063  0.818
I(U_J^2)     1 0.011740 0.011740   1.502  0.308
T1_J         1 0.000730 0.000730   0.093  0.780
T2_J         1 0.000001 0.000001   0.000  0.991
U1_I         1 0.012678 0.012678   1.622  0.293
U_I          1 0.000421 0.000421   0.054  0.831
U_J          1 0.000035 0.000035   0.004  0.951
VEC1         1 0.003009 0.003009   0.385  0.579
VEC2         1 0.000038 0.000038   0.005  0.949
I(T1_J^3)    1 0.000923 0.000923   0.118  0.754
I(T2_J^3)    1 0.000045 0.000045   0.006  0.944
I(U1_I^3)    1 0.009060 0.009060   1.159  0.361
I(U_I^3)     1 0.000015 0.000015   0.002  0.967
I(U_J^3)     1 0.004588 0.004588   0.587  0.499
Residuals    3 0.023451 0.007817
```

```
Using Model: . ~ .
Prediction Regression Step:            Df   Sum Sq  Mean Sq F value Pr(>F)
I(T1_J^2)    1 0.000005 0.000005   0.001  0.982
I(T2_J^2)    1 0.001378 0.001378   0.176  0.703
I(U1_I^2)    1 0.000101 0.000101   0.013  0.917
I(U_I^2)     1 0.000493 0.000493   0.063  0.818
I(U_J^2)     1 0.011740 0.011740   1.502  0.308
T1_J         1 0.000730 0.000730   0.093  0.780
T2_J         1 0.000001 0.000001   0.000  0.991
U1_I         1 0.012678 0.012678   1.622  0.293
U_I          1 0.000421 0.000421   0.054  0.831
U_J          1 0.000035 0.000035   0.004  0.951
VEC1         1 0.003009 0.003009   0.385  0.579
VEC2         1 0.000038 0.000038   0.005  0.949
I(T1_J^3)    1 0.000923 0.000923   0.118  0.754
I(T2_J^3)    1 0.000045 0.000045   0.006  0.944
I(U1_I^3)    1 0.009060 0.009060   1.159  0.361
I(U_I^3)     1 0.000015 0.000015   0.002  0.967
I(U_J^3)     1 0.004588 0.004588   0.587  0.499
Residuals    3 0.023451 0.007817
```