Suggestions for multi-fidelity R-SPLINE

Graham Burgess

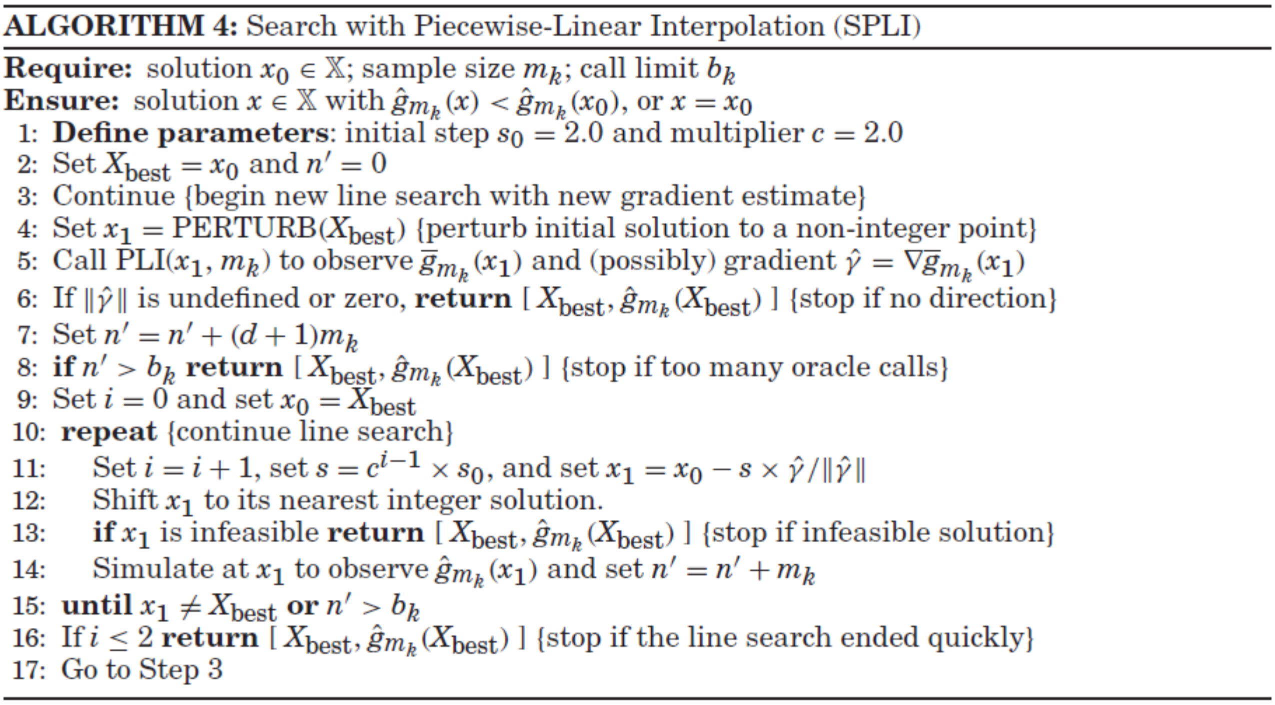
December 2024

The retrospective search with piecewise-linear interpolation and neighborhood enumeration, R-SPLINE (Wang et al., 2013), is a sample-average approximation procedure for discrete problems. Each iteration of R-SPLINE solves a sample-path problem retrospectively (R) using the SPLINE procedure. It is retrospective in the sense that each iteration of SPLINE uses the solution from the previous iteration as a 'warm-start'. The SPLINE procedure repeatedly performs a search with piecewise linear interpolation (SPLI) and a neighbourhood enumeration (NE). The SPLI procedure performs a series of line searches. Each line search begins with gradient estimation at a point on the continuous domain, using piecewise linear interpolation (PLI). The SPLI procedure ends with a new point on the integer domain, along with a estimation of its objective value (estimated using multiple simulation replications). Figure 1 is a screenshot of the SPLI procedure from the original R-SPLINE paper (Wang et al., 2013).

PLI estimates the gradient at a point in the continuous domain by estimating the objective value at all $d + 1$ integer points on the surrounding simplex. Figure 2 is a screenshot of the PLI procedure from the original R-SPLINE paper (Wang et al., 2013).The gradient in a given co-ordinate direction is the difference between estimated objective values at opposite sides of the simplex, in that co-ordinate direction. Gradients are normalised when used in the line search. Objective value estimation at integer points, performed in line 8 of the PLI procedure, is done with multiple simulation replications, which could be computationally expensive. Our main suggestion is simple - in this part of the PLI procedure, we replace the multiple simulation replications with a single run of a low-fidelity model. In the homeless care system (Burgess et al., 2024), this low-fidelity model could be a fluid model.

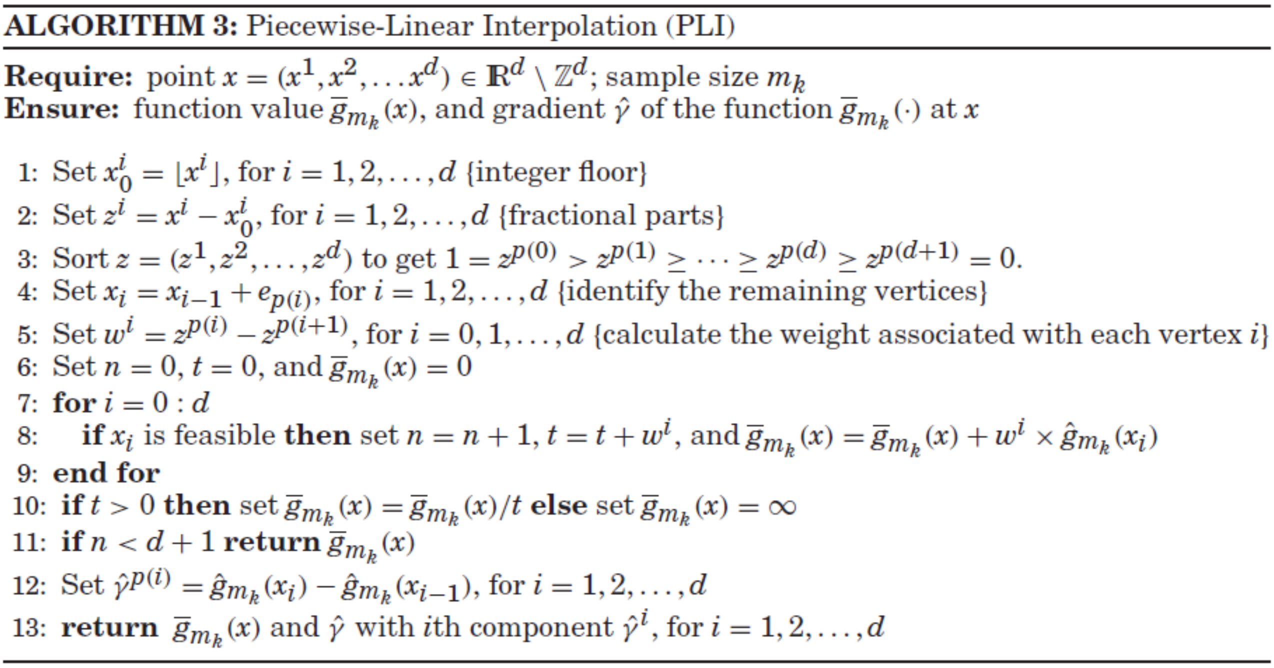
We propose the following in case bias in the low-fidelity model leads to poor gradient estimates:

- At the start of the R-SPLINE procedure, once an initial solution has been set, we could fit a meta model of the objective function response using some initial simulation effort. We could consider a local or a global meta-model. A local model could have the advantage of performing well locally in a focused manner, using for example linear or polynomial terms. The disadvantage of this would be that as we moved around the solution space we would need to regularly refit this model with more simulation effort. A global model may not perform as well locally but may need less simulation effort to maintain throughout the procedure. We could consider a meta-model on the integer-ordered or continuous-valued domain. The former may be more suitable for R-SPLINE (an integer-ordered method) but continuous models such as polynomials or Gaussian process models, may be easier to fit. An appropriate meta model may consist of a 'physical' term using a low-fidelity



```
ALGORITHM 4: Search with Piecewise-Linear Interpolation (SPLI)
Require: solution $x_0 \in \mathbb{X}$; sample size $m_k$; call limit $b_k$
Ensure: solution $x \in \mathbb{X}$ with $\hat{g}_{m_k}(x) < \hat{g}_{m_k}(x_0)$, or $x = x_0$
 1: Define parameters: initial step $s_0 = 2.0$ and multiplier $c = 2.0$
 2: Set $X_{\text{best}} = x_0$ and $n' = 0$
 3: Continue {begin new line search with new gradient estimate}
 4: Set $x_1 = \text{PERTURB}(X_{\text{best}})$ {perturb initial solution to a non-integer point}
 5: Call PLI($x_1, m_k$) to observe $\overline{g}_{m_k}(x_1)$ and (possibly) gradient $\hat{\gamma} = \nabla\overline{g}_{m_k}(x_1)$
 6: If $\|\hat{\gamma}\|$ is undefined or zero, return [ $X_{\text{best}}, \hat{g}_{m_k}(X_{\text{best}})$ ] {stop if no direction}
 7: Set $n' = n' + (d+1)m_k$
 8: if $n' > b_k$ return [ $X_{\text{best}}, \hat{g}_{m_k}(X_{\text{best}})$ ] {stop if too many oracle calls}
 9: Set $i = 0$ and set $x_0 = X_{\text{best}}$
10: repeat {continue line search}
11:     Set $i = i + 1$, set $s = c^{i-1} \times s_0$, and set $x_1 = x_0 - s \times \hat{\gamma}/\|\hat{\gamma}\|$
12:     Shift $x_1$ to its nearest integer solution.
13:     if $x_1$ is infeasible return [ $X_{\text{best}}, \hat{g}_{m_k}(X_{\text{best}})$ ] {stop if infeasible solution}
14:     Simulate at $x_1$ to observe $\hat{g}_{m_k}(x_1)$ and set $n' = n' + m_k$
15: until $x_1 \neq X_{\text{best}}$ or $n' > b_k$
16: If $i \leq 2$ return [ $X_{\text{best}}, \hat{g}_{m_k}(X_{\text{best}})$ ] {stop if the line search ended quickly}
17: Go to Step 3
```

Figure 1: SPLI procedure (screenshot from original paper)



```
ALGORITHM 3: Piecewise-Linear Interpolation (PLI)
Require: point $x = (x^1, x^2, \dots x^d) \in \mathbb{R}^d \setminus \mathbb{Z}^d$; sample size $m_k$
Ensure: function value $\overline{g}_{m_k}(x)$, and gradient $\hat{\gamma}$ of the function $\overline{g}_{m_k}(\cdot)$ at $x$

 1: Set $x_0^i = \lfloor x^i \rfloor$, for $i = 1, 2, \dots, d$ {integer floor}
 2: Set $z^i = x^i - x_0^i$, for $i = 1, 2, \dots, d$ {fractional parts}
 3: Sort $z = (z^1, z^2, \dots, z^d)$ to get $1 = z^{p(0)} > z^{p(1)} \geq \dots \geq z^{p(d)} \geq z^{p(d+1)} = 0$.
 4: Set $x_i = x_{i-1} + e_{p(i)}$, for $i = 1, 2, \dots, d$ {identify the remaining vertices}
 5: Set $w^i = z^{p(i)} - z^{p(i+1)}$, for $i = 0, 1, \dots, d$ {calculate the weight associated with each vertex $i$}
 6: Set $n = 0$, $t = 0$, and $\overline{g}_{m_k}(x) = 0$
 7: for $i = 0 : d$
 8:     if $x_i$ is feasible then set $n = n + 1$, $t = t + w^i$, and $\overline{g}_{m_k}(x) = \overline{g}_{m_k}(x) + w^i \times \hat{g}_{m_k}(x_i)$
 9: end for
10: if $t > 0$ then set $\overline{g}_{m_k}(x) = \overline{g}_{m_k}(x)/t$ else set $\overline{g}_{m_k}(x) = \infty$
11: if $n < d + 1$ return $\overline{g}_{m_k}(x)$
12: Set $\hat{\gamma}^{p(i)} = \hat{g}_{m_k}(x_i) - \hat{g}_{m_k}(x_{i-1})$, for $i = 1, 2, \dots, d$
13: return $\overline{g}_{m_k}(x)$ and $\hat{\gamma}$ with $i$th component $\hat{\gamma}^i$, for $i = 1, 2, \dots, d$
```

Figure 2: PLI procedure (screenshot from original paper)

model and a corrective term. Osorio and Chong (2015) use this idea for a global meta model which weights information from all points on the solution space depending on the distance between those points and the point of searching. As the search continues, the weights change.

- We could use our meta model in PLI (line 8) for objective value estimation at integer points.
- During the line search in SPLI (line 14), simulation effort is used to estimate the objective value at new integer points. We could use this information to update or validate our chosen meta model.
- For the first gradient estimation in each SPLI run (first time at line 5), we could obtain two gradient estimates - one using simulation (as in the original R-SPLINE paper) and one with the meta model (as we propose). We could evaluate the quality of the meta model-based gradient estimates and only proceed with it in the given SPLI run if the quality was sufficient.
- Another way of validating the gradient information from the meta-model could be to compare the actual change in objetive value with each new solution, to the predicted change from the meta-model based gradient information.

References

Burgess, G., Singham, D. I., and Rhodes-Leader, L. (2024). Time-varying capacity planning for designing large-scale homeless care systems. *Under Review*.

Osorio, C. and Chong, L. (2015). A computationally efficient simulation-based optimization algorithm for large-scale urban transportation problems. *Transportation Science*, 49(3):623–636.

Wang, H., Pasupathy, R., and Schmeiser, B. W. (2013). Integer-ordered simulation optimization using r-spline: Retrospective search with piecewise-linear interpolation and neighborhood enumeration. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 23(3):1–24.