A. Guarantees with known (ν_*, ρ_*)

In this section we prove that if Algorithm 1 is run with the parameters (ν_*, ρ_*), then it terminates with an x_Λ that is close to optimal.

Recall that $\nu > \nu_*$ and $\rho > \rho_*$. Consider a cell $\mathcal{P}_{h,i_h^*}$ at height h such that $x^* \in \mathcal{P}_{h,i_h^*}$. From Assumption 1 we have that:

$$\begin{aligned} b_{h,i_h^*} &= f_{z_h}(x_{h,i_h^*}) + \zeta_(z_h) + \nu\rho^h \\ &\geq f(x_{h,i_h^*}) + \nu\rho^h \geq f^* \end{aligned} \tag{5}$$

Therefore, any node (h,i) such that $b_{h,i} < f^*$ will never be expanded. Therefore, the nodes at height h that are expanded form a subset of G_h defined below:

$$G_h \triangleq \{\text{nodes } (h,i) \text{ such that } f_{z_h}(x_{h,i}) + \nu\rho^h + \zeta_{z_h}(x_{h,i}) \geq f^*\}.$$

By definition of z_h we have that,

$$G_h \subseteq \left\{\text{nodes } (h,i) \text{ such that } f_{z_h}(x_{h,i}) + 2\nu\rho^h \geq f^*\right\}.$$

Therefore, by Assumption 1 and Definition 1 we have the following lemma.

Lemma 1. *We have* $|G_h| \leq C(\nu,\rho)\rho^{-d(\nu,\rho)h}$.

We now argue that the tree has to grow to a certain minimum depth given a cost budget Λ in Algorithm 1.

Lemma 2. *Let h' be the biggest number h such $\sum_{l=0}^{h} C(\nu,\rho)K\lambda(z_l)\rho^{-d(\nu,\rho)l} \leq \Lambda$. The tree in Algorithm 1 grows to a height of at least $h(\Lambda) = h'+1$, and uses a cost budget of at most $\Lambda + K\lambda(1)$ when it terminates.*

Proof. We have shown that only the nodes in $G = \cup_h G_h$ are expanded. Let us consider the strategy that only expands nodes in G, but expands the leaf among the current leaves with the least height. This strategy yields the tree with minimum height among strategies that only expand nodes in G. The cost incurred by this strategy till step h' is given by,

$$\sum_{l=0}^{h'} C(\nu,\rho)K\lambda(z_l)\rho^{-d(\nu,\rho)l} \leq \Lambda.$$

Since the above cost is less than or equal to Λ another set of children at height $h'+1$ is expanded and then the algorithm terminates because of the check in the while loop in step 4 of Algorithm 1. Therefore, the resultant tree has a height of at least $h'+1$ and incurs a cost budget of at most $\Lambda + K\lambda(1)$. ∎

Proof of Theorem 1. The proof of Theorem 1 follows naturally from Lemma 2 and the definition of G_h. Since, a node point $x_{h'+1,j}$ at height $h(\Lambda) = h'+1$ has been evaluated, it means that $x_{h'+1,j} \in G_{h'+1}$. Therefore, we have that

$$f(x_{h(\Lambda),j}) \geq f^* - 2\nu\rho^{h(\Lambda)}. \tag{6}$$

∎

Now we prove Corollary 1 under Assumptions 2 and 3 separately.

Proof of Corollary 1. Consider Algorithm 1 with parameters (ν,ρ).

(i) Under Assumption 2: Note that $\lambda(z_h) \leq \beta h$. Therefore, we have the following chain,

$$\begin{aligned} \sum_{l=0}^{h} \lambda(z_l)\rho^{-d(\nu,\rho)l} &\leq \sum_{l=0}^{h} \beta l \rho^{-d(\nu,\rho)l} \\ &\leq \beta \frac{h\rho^{-d(\nu,\rho)(h+1)}}{\rho^{-d(\nu,\rho)} - 1} \end{aligned}$$

Therefore, from Theorem 1 we have the following,

$$\Lambda \leq C(\nu,\rho)K\beta \frac{h(\Lambda)\rho^{-d(\nu,\rho)(h(\Lambda)+1)}}{\rho^{-d(\nu,\rho)} - 1}.$$

Suppose Λ is large enough such that $h(\Lambda) \leq \rho^{-\epsilon h(\Lambda)}$ where ϵ is a small constant. Then we have the following:

$$\begin{aligned} R_\Lambda &\leq 2\nu\rho^{h(\Lambda)} \\ &\leq 2\nu \left(\frac{C(\nu,\rho)K\beta}{\Lambda(1 - \rho^{d(\nu,\rho)})} \right)^{\frac{1}{d(\nu,\rho)+\epsilon}} \end{aligned}$$

(i) Under Assumption 3: Note that $\lambda(z_h) \leq \gamma^{-h}$. Therefore, we have the following,

$$\begin{aligned} \sum_{l=0}^{h} \lambda(z_l)\rho^{-d(\nu,\rho)l} &\leq \frac{\gamma^{-(h+1)}\rho^{-d(\nu,\rho)(h+1)} - 1}{\gamma^{-1}\rho^{-d(\nu,\rho)} - 1} \\ &\leq \frac{\rho^{-(d(\nu,\rho)+1)(h+1)}}{\gamma^{-1}\rho^{-d(\nu,\rho)} - 1} \end{aligned}$$

Therefore, we have that,

$$\begin{aligned} R_\Lambda &\leq 2\nu\rho^{h(\Lambda)} \\ &\leq 2\frac{\nu}{\rho} \left(\frac{2C(\nu,\rho)K}{\Lambda(\gamma^{-1}\rho^{-d(\nu,\rho)} - 1)} \right)^{\frac{1}{d(\nu,\rho)+1}} \end{aligned}$$

∎

B. Recovering optimal scaling with unknown smoothness

In this section, we relate the optimality dimension $d(\nu, \rho)$ to $d(\nu_*, \rho_*)$ for $\nu > \nu_*$ and $\rho > \rho_*$. These relations are implied by the analysis of Theorem 1 in (Grill et al., 2015).

Lemma 3. *Consider the parameters $\nu > \nu_*$ and $\rho > \rho_*$. Let $h_{min} \triangleq \log(\nu/\nu_*)\log(1/\rho)$. Then we have the following,*

$$\mathcal{N}_h(2\nu\rho^h) \\ \leq \max\left(C(\nu_*, \rho_*)K^{(\log \rho_* + \log \nu_* - \log \nu)/\log \rho}, K^{h_{min}}\right) \times \\ \rho^{-h[d(\nu_*,\rho_*) + \log K(1/\log(1/\rho) - 1/\log(1/\rho_*))]}$$

Proof. It follows directly from the analysis of Theorem 1 in appendix B.1 of (Grill et al., 2015). □

Lemma 3 implies the following,

$$C(\nu, \rho) \leq \max\left(C(\nu_*, \rho_*)K^{(\log \rho_* + \log(\nu_*/\nu))/\log \rho}, K^{h_{min}}\right) \\ d(\nu, \rho) \leq d(\nu_*, \rho_*) + \log K(1/\log(1/\rho) - 1/\log(1/\rho_*)) \tag{7}$$

C. Putting it together: Simple Regret Bound

Let $R^{\nu,\rho}_{\Lambda_0}$ be the simple regret of Algorithm 1 with parameters ν, ρ. Note that Algorithm 2 is designed such that its simple regret is equal to at most the simple regret of one of the MFDOO instances spawned. We will analyze Algorithm 2 under Assumptions 2 and 3 separately.

Proof of Theorem 2. The proof is divided into two sections corresponding to Assumptions 2 and 3 respectively. Consider $\rho \geq \rho_*$ and $\nu \geq \nu_*$. In this analysis we assume $d(\nu_*, \rho_*) > 0$.

Under Assumption 2: We have the following chain,

$$\log R^{\nu,\rho}_{\Lambda_0} \leq \log(2\nu) + \frac{\log C(\nu, \rho)}{d(\nu, \rho) + \epsilon} + \frac{\log(K\beta)}{d(\nu, \rho) + \epsilon} \\ + \frac{\log(1/(1 - \rho^{d(\nu,\rho)}))}{d(\nu, \rho) + \epsilon} - \frac{\log \Lambda_0}{d(\nu, \rho) + \epsilon} \\ \leq \log(2\nu_{max}) + \frac{\log C(\nu, \rho)}{d(\nu, \rho) + \epsilon} + \frac{\log(K\beta)}{d(\nu_*, \rho_*) + \epsilon} \\ + \frac{\log(1/(1 - \rho^{d(\nu_*,\rho_*)}))}{d(\nu_*, \rho_*) + \epsilon} \\ - \frac{\log \Lambda_0}{d(\nu_*, \rho_*) + \epsilon}\left(1 - \frac{d(\nu, \rho) - d(\nu_*, \rho_*)}{2 + d(\nu_*, \rho_*)}\right)$$

Let $\rho_i = \rho_{max}^{N/i}$ for $i \in \{1, 2, ..., N\}$. We define,

$$\bar{\rho} \triangleq \underset{i:\rho_i \geq \rho_*}{\text{argmin}}[d(\nu_{max}, \rho_i) - d(\nu_*, \rho_*)]$$

Note that $\bar{\rho}$ is the best $\rho_i \geq \rho_*$ that is spawned as a MF-DOO instance in Algorithm 2. Thus bounding the regret of $R^{\nu_{max},\bar{\rho}}_{\Lambda_0}$ for $\Lambda_0 = \Lambda/N - \lambda(1)$ immediately yields a simple regret bound for Algorithm 2. Now we observe that,

$$d(\nu_{max}, \bar{\rho}) - d(\nu_*, \rho_*) \leq \frac{D_{max}}{N}.$$

Therefore, we have the following,

$$\log R^{\nu_{max},\bar{\rho}}_{\Lambda_0} \leq \log(2\nu_{max}) + \frac{\log C(\nu_{max}, \bar{\rho})}{d(\nu_{max}, \bar{\rho}) + \epsilon} + \frac{\log(K\beta)}{d(\nu_*, \rho_*) + \epsilon} \\ + \frac{\log(1/(1 - \bar{\rho}^{d(\nu_*,\rho_*)}))}{d(\nu_*, \rho_*) + \epsilon} \\ - \log \Lambda_0\left(\frac{1}{d(\nu_*, \rho_*) + \epsilon} - \frac{D_{max}/N}{(\epsilon + d(\nu_*, \rho_*))^2}\right)$$

We can bound the second term as follows,

$$\frac{\log C(\nu_{max}, \bar{\rho})}{d(\nu_{max}, \bar{\rho}) + \epsilon} \leq \frac{\log C(\nu_{max}, \bar{\rho})}{d(\nu_*, \rho_*) + \epsilon} \\ \leq \frac{1}{d(\nu_*, \rho_*) + \epsilon}\log\max\left(C(\nu_*, \rho_*)K^{(\log \rho_* + \log(\nu_*/\nu))/\log \rho}, K^{h_{min}}\right) \\ \leq a + \frac{D_{max}}{d(\nu_*, \rho_*) + \epsilon}\log(\nu_{max}/\nu_*),$$

where a is a constant independent of all the parameters. Finally we can bound the last term as follows,

$$\log \Lambda_0\left(-\frac{1}{d(\nu_*, \rho_*) + \epsilon} + \frac{D_{max}/N}{(\epsilon + d(\nu_*, \rho_*))^2}\right) \\ \leq -\frac{\log \Lambda_0}{d(\nu_*, \rho_*) + \epsilon} + \log \Lambda_0 \frac{2}{\log(\Lambda/\log \Lambda)}\frac{1}{(\epsilon + d(\nu_*, \rho_*))^2} \\ \leq -\frac{\log \Lambda_0}{d(\nu_*, \rho_*) + \epsilon} + \frac{2}{(\epsilon + d(\nu_*, \rho_*))^2}$$

where the second inequality follows from the definition of N. Now, we can finally bound the regret of Algorithm 2 as follows:

$$R^{\nu_{max},\bar{\rho}}_{\Lambda_0} \\ \leq 2\nu_{max}\exp\left(a + \frac{2}{(\epsilon + d(\nu_*, \rho_*))^2}\right)(\nu_{max}/\nu_*)^{\frac{D_{max}}{\epsilon + d(\nu_*,\rho_*)}} \times \\ (K\beta/(1 - \bar{\rho}^{d(\nu_*,\rho_*)}))^{1/(\epsilon + d(\nu_*,\rho_*))}\Lambda_0^{-\frac{1}{\epsilon + d(\nu_*,\rho_*)}} \\ = \mathcal{O}\left((\nu_{max}/\nu_*)^{\frac{D_{max}}{\epsilon + d(\nu_*,\rho_*)}} \times \\ \left(\frac{2\Lambda}{K\beta D_{max}\log(\Lambda/\log \Lambda)} - \frac{\lambda(1)}{K\beta}\right)^{-\frac{1}{\epsilon + d(\nu_*,\rho_*)}}\right)$$

Under Assumption 3: Now we prove similar results under the second assumption on the cost and bias function. The analysis is very similar to the first part of the theorem. Note that $\gamma > \rho_{max}$. We follow the same notational convention as the first part of the theorem. Proceeding exactly as above, we have the following chain,

$$\begin{aligned}
&\log R^{\nu_{max},\bar{\rho}}_{\Lambda_0} \leq \log(2\nu_{max}/\rho_*) \\
&+ \frac{\log 2C(\nu_{max},\bar{\rho})}{d(\nu_{max},\bar{\rho})+1} + \frac{\log K}{d(\nu_*,\rho_*)+1} - \frac{\log(\gamma^{-1}\bar{\rho}^{d(\nu_*,\rho_*)}-1)}{d(\nu_*,\rho_*)+1} \\
&- \log \Lambda_0 \left(\frac{1}{d(\nu_*,\rho_*)+1} - \frac{D_{max}/N}{(1+d(\nu_*,\rho_*))^2} \right) \\
&\leq \log(2\nu_{max}/\rho_*) + 2a + \frac{2D_{max}}{d(\nu_*,\rho_*)+1} \log(\nu_{max}/\nu_*) \\
&+ \frac{\log K}{d(\nu_*,\rho_*)+1} - \frac{\log(\gamma^{-1}\bar{\rho}^{d(\nu_*,\rho_*)}-1)}{d(\nu_*,\rho_*)+1} \\
&- \frac{\log \Lambda_0}{d(\nu_*,\rho_*)+1} + 4
\end{aligned}$$

Thus we get the following regret bound:

$$\begin{aligned}
&R^{\nu_{max},\bar{\rho}}_{\Lambda_0} \leq 2(\nu_{max}/\rho_*) \exp(2a+4) (\nu_{max}/\nu_*)^{\frac{2D_{max}}{1+d(\nu_*,\rho_*)}} \\
&\times \left(\frac{1}{\gamma^{-1}\bar{\rho}^{d(\nu_*,\rho_*)-1}} \right)^{1/(1+d(\nu_*,\rho_*))} \Lambda_0^{-\frac{1}{1+d(\nu_*,\rho_*)}} \\
&= \mathcal{O}\Bigg((\nu_{max}/\nu_*)^{\frac{2D_{max}}{1+d(\nu_*,\rho_*)}} \times \\
&\left(\frac{2\Lambda}{K D_{max} \log(\Lambda/\log\Lambda)} - \frac{\lambda(1)}{K} \right)^{-\frac{1}{1+d(\nu_*,\rho_*)}} \Bigg)
\end{aligned}$$

□

D. Description of Synthetic Functions

The following are the synthetic functions used in the paper (Currin, 1988; Dixon & Szego, 1978).

Currin exponential function (Currin, 1988): The domain is the two dimensional unit cube $\mathcal{X} = [0,1]^2$ and the fidelity is $\mathcal{Z} = [0,1]$. We used $\lambda(z) = 0.1 + z^2$, $\sigma^2 = 0.5$ and,

$$f_z(x) = \left(1 - 0.1(1-z)\exp\left(\frac{-1}{2x_2}\right)\right) \left(\frac{2300x_1^3 + 1900x_1^2 + 2092x_1 + 60}{100x_1^3 + 500x_1^2 + 4x_1 + 20} \right).$$

Hartmann functions (Dixon & Szego, 1978): We used $f_z(x) = \sum_{i=1}^{4} (\alpha_i - \alpha'(z)) \exp\left(-\sum_{j=1}^{3} A_{ij}(x_j - P_{ij})^2\right)$. Here A, P are given below for the 3 and 6 dimensional cases and $\alpha = [1.0, 1.2, 3.0, 3.2]$. Then α' was set as $\alpha'(z) = 0.1(1-z)$. We constructed the $p = 4$ and $p = 2$ Hartmann functions for the 3 and 6 dimensional cases respectively this way. When $z = 1$, this reduces to the usual Hartmann function commonly used as a benchmark in global optimisation.

For the 3 dimensional case we used $\lambda(z) = 0.05 + (1 - 0.05)z^3$, $\sigma^2 = 0.01$ and,

$$A = \begin{bmatrix} 3 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3 & 10 & 30 \\ 0.1 & 10 & 35 \end{bmatrix}, \quad P = 10^{-4} \times \begin{bmatrix} 3689 & 1170 & 2673 \\ 4699 & 4387 & 7470 \\ 1091 & 8732 & 5547 \\ 381 & 5743 & 8828 \end{bmatrix}.$$

For the 6 dimensional case we used $\lambda(z) = 0.05 + (1 - 0.05)z^3$, $\sigma^2 = 0.05$ and,

$$A = \begin{bmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{bmatrix},$$

$$P = 10^{-4} \times \begin{bmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{bmatrix}.$$

Branin function (Dixon & Szego, 1978): We use the following function where $\mathcal{X} = [[-5,10],[0,15]]^2$ and $\mathcal{Z} = [0,1]$.

$$f_z(x) = a(x_2 - b(z)x_1^2 + c(z)x_1 - r)^2 + s(1-t(z))\cos(x_1) + s,$$

where $a = 1$, $b(z) = 5.1/(4\pi^2) - 0.01(1-z)$ $c(z) = 5/\pi - 0.1(1-z)$, $r = 6$, $s = 10$ and $t(z) = 1/(8\pi) + 0.05(1-z)$. At $z = 1$, this becomes the standard Branin function used as a benchmark in global optimization. We used $\lambda(z) = 0.05 + z^3$ for the cost function and $\sigma^2 = 0.05$ for the noise variance.