Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

REGULAR TREE SEARCH FOR SIMULATION OPTIMIZATION

Du-Yi Wang[1,2]

[1]Dept. of Decision Analytics and Operations, College of Business, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong SAR, CHINA
[2]Institute of Statistics and Big Data, Renmin University of China, Beijing, CHINA

ABSTRACT

Tackling simulation optimization problems with non-convex objective functions remains a fundamental challenge in operations research. In this paper, we propose a class of random search algorithms, called Regular Tree Search, which integrates adaptive sampling with recursive partitioning of the search space. The algorithm concentrates simulations on increasingly promising regions by iteratively refining a tree structure. A tree search strategy guides sampling decisions, while partitioning is triggered when the number of samples in a leaf node exceeds a threshold that depends on its depth. Furthermore, a specific tree search strategy, Upper Confidence Bounds applied to Trees, is employed in the Regular Tree Search. We prove global convergence under sub-Gaussian noise, based on assumptions involving the optimality gap, without requiring continuity of the objective function. Numerical experiments confirm that the algorithm reliably identifies the global optimum and provides accurate estimates of its objective value.

1 ALGORITHM

The Regular Tree Search integrates two core mechanisms: adaptive space partitioning (via a tree) and exploration-exploitation balancing (via a tree search strategy), and operates under a two-stage workflow.

1.1 Regular Tree

The algorithm first constructs a Regular Tree to partition the feasible domain ($\mathcal{X} \subseteq \mathbb{R}^d$, canonicalized as $[0,1]^d$). The tree enforces four key properties to ensure split quality: (1) **Honesty**: Disjoint datasets for leaf estimation ($\mathbb{S}_{\mathcal{I}}$) and split decisions ($\mathbb{S}_{\mathcal{J}}$) avoid data leakage; (2) **Random-Split**: Each feature has at least κ/d probability of being split, preventing dimension neglect; (3) α**-Spatial Balance**: Child nodes retain at least α-fraction of the parent's size; (4) $f(c)$**-Sample Balance**: Depth-c leaves have $\mathbb{S}_{\mathcal{I}}$ samples bounded by $f(c)$ (e.g., $c \log c$) for reliable estimation.

1.2 Regular Tree Search

The algorithm proceeds in two sequential stages, followed by a termination step to output the solution:

Stage 1: Initialization (Warm-Up). This stage establishes an initial partition of the feasible domain to guide subsequent adaptive sampling:

1. Allocate an initial sample budget $N_0 < N$ (total budget). Uniformly sample N_0 points from $\mathcal{X}$, and run one simulation per point to generate a dataset $\mathbb{S}$.
2. Split $\mathbb{S}$ into two disjoint subsets: $\mathbb{S}_{\mathcal{I}}$ (size $\lfloor N_0/2 \rfloor$, for leaf value estimation) and $\mathbb{S}_{\mathcal{J}}$ (size $N_0 - \lfloor N_0/2 \rfloor$, for split decisions). Using the splitting criterion that enforces the four Regular Tree properties above, construct the initial Regular Tree.

Stage 2: Adaptive Sampling and Tree Refinement. This iterative stage refines the tree and focuses simulations on promising regions, balancing exploration and exploitation via the *UCT* strategy:

1. *Leaf Selection*: For each iteration, select a leaf node using UCT—this strategy balances "exploiting" leaves with high sample means (likely promising regions) and "exploring" under-sampled leaves (uncertain regions) via the formula: $\text{ucb}_l = \bar{Y}_{L_l} + C_p \sqrt{\frac{2 \log n_p}{n_l}}$. Here, $\bar{Y}_{L_l}$ is the sample mean of

Table 1: Statistics of the true function value of the optimal solution estimate of four algorithms.

					Quantile of value			
d	Algorithm	Mean	RMSE	Best	25%	50%	75%	Worst
2	Regular Tree Search	3.20	4.17	0.03	1.28	2.11	4.99	10.23
	ASR	5.23	6.16	0.31	3.02	4.78	6.73	20.50
	IHR-SO	36.54	36.74	22.61	34.24	37.18	39.69	43.80
	AP-SO	35.77	36.16	21.06	33.52	37.06	39.77	43.83

leaf L_l (from $\mathbb{S}_\mathcal{I}$), n_l is the number of $\mathbb{S}_\mathcal{I}$ samples in L_l, n_p is the $\mathbb{S}_\mathcal{I}$ sample count of L_l's parent node, and C_p is a hyperparameter tuning the exploration-exploitation trade-off.

2. *New Sampling*: Uniformly sample a new point from the selected leaf, run a simulation, and add the resulting $(\mathbf{x}, Y)$ pair to $\mathbb{S}_\mathcal{I}$.
3. *Tree Refinement*: If the number of $\mathbb{S}_\mathcal{I}$ samples in the selected leaf reaches $f(c)$ (where c is the leaf's depth), supplement $\mathbb{S}_\mathcal{J}$ with additional samples (until its count in the leaf also meets $f(c)$). Split the leaf into two child nodes using the Regular Tree criterion, then update the set of active leaf nodes by replacing the parent leaf with its two children.

When the total sample budget N is exhausted, identify the leaf node with the largest $\mathbb{S}_\mathcal{I}$ sample mean (the most promising region). The midpoint of this leaf is the estimated optimal solution, and its sample mean is the estimated optimal objective value.

2 THEORETICAL CONVERGENCE

Two core convergence results of Regular Tree Search are formally proven under three key theoretical assumptions: (1) Sub-Gaussian simulation noise (i.e., noise meeting sub-Gaussian moment conditions), (2) Local smoothness around the unique optimal point $\mathbf{x}^*$, and (3) A local optimality gap (weaker than the objective function's global continuity). Regular Tree Search also requires additional constraints on tree structure and sampling strategy: each leaf node's depth is bounded between $h^-(N)$ and $h^+(N)$, with $\lim_{N\to\infty} h^-(N) = \infty$ and $\lim_{N\to\infty} \frac{h^+(N)}{f(h^-(N))} = 0$. These results are detailed below:

1. **Spatial Convergence**: As total sample budget $N \to \infty$, the distance between the selected optimal leaf node L_N^* and true optimal point $\mathbf{x}^*$ converges to 0 in probability.

2. **Value Convergence**: The sample mean of optimal leaf node L_N^* (from $\mathbb{S}_\mathcal{I}$) converges to the true optimal objective value $\mu(\mathbf{x}^*)$ in probability.

Critically, these guarantees do not depend on the objective function's global continuity—they rely solely on $\mathbf{x}^*$'s local properties. This distinction is pivotal, as it greatly broadens the algorithm's applicability to real-world scenarios where objective functions are often discontinuous or lack global smoothness.

3 NUMERICAL EXPERIMENTS

We compare Regular Tree Search against three comparative algorithms (ASR, IHR-SO, AP-SO) on non-convex simulation problems. Consider the 2-dimensional Rastrigin function: $\mu(\mathbf{x}) = 20 + \sum_{i=1}^{2}[x_i^2 - 10\cos(2\pi x_i)] \quad (\mathbf{x} = (x_1, x_2))$, with feasible domain $\mathcal{X} = [-5, 5]^2$. It aims to find the global minimum at $\mathbf{x}^* = (0, 0)$ (where $\mu(\mathbf{x}^*) = 0$) and has multiple local minima. Let $\epsilon(\mathbf{x}) \sim N(0, 1)$; for illustration, we convert the original maximization problem (consistent with our algorithm design) to minimization, equivalent to maximizing the negative Rastrigin function. All algorithms are evaluated under a fixed total sample budget of 1000. Table 1 shows that the proposed Regular Tree Search outperforms other algorithms: it achieves the lowest mean error and RMSE, and its results are more concentrated.