Batched GPU Execution for High-Throughput Combinatorial Search

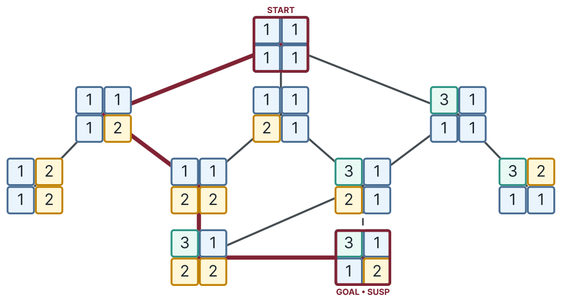
Minh Phuc Nguyen (Author) · Matthew Anderson (Advisor)
Union College, Schenectady, New York, USA

CPU-GPU Synchronization Stalls Search

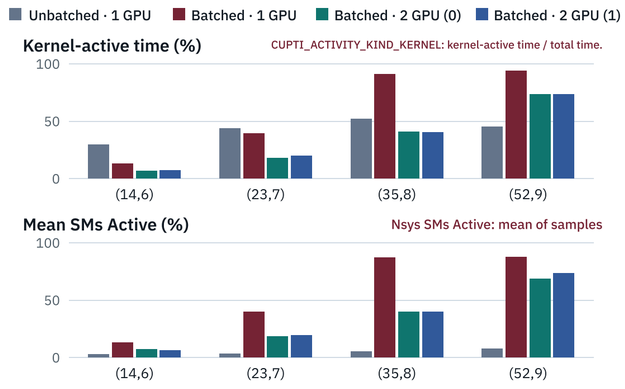
Motivation: Strong Uniquely Solvable Puzzles (SUSPs) are arrays over {1,2,3}. An (s,k)-puzzle has s distinct rows and k columns. For simplifiable SUSPs, more rows at fixed width give tighter bounds on how matrix-multiplication cost grows [1,2]. Finding larger puzzles requires many candidate evaluations.

The problem: Per-candidate GPU offloading makes CPU search workers wait for evaluation results.

Main Contribution: A batched CPU–GPU search pipeline that decouples frontier expansion from candidate evaluation through a work queue and reusable buffers.



Profiling Reveals GPU Activity and Headroom

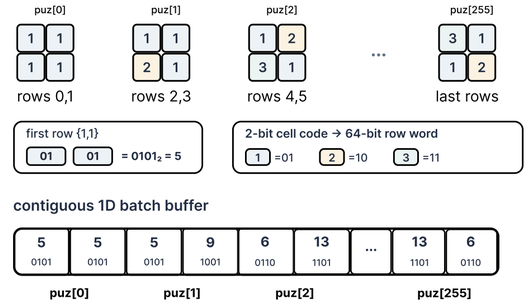


Unbatched Search exposes too little parallel work, leaving many SMs idle even as puzzle size and computation increase. Batched Search evaluates candidates concurrently, allowing larger puzzles to increase computation and keep the GPU busier. We hypothesize that queued batches keep one GPU supplied with work, whereas distributing batches across two GPUs creates more idle gaps per device, lowering average activity while parallel execution still increases total throughput.

Asynchronous Queueing Decouples CPU and GPU Execution

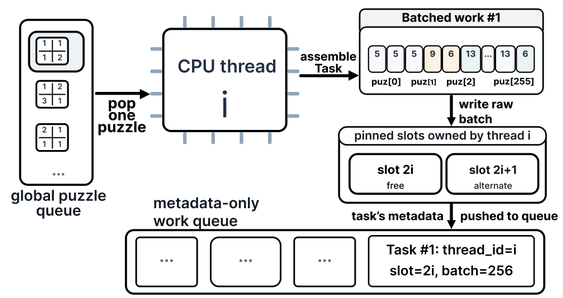
Packed Rows Form Contiguous GPU Batches ①

Each cell can be {1, 2, 3}, so 2 bits encode one cell; each packed row is a 64-bit integer.



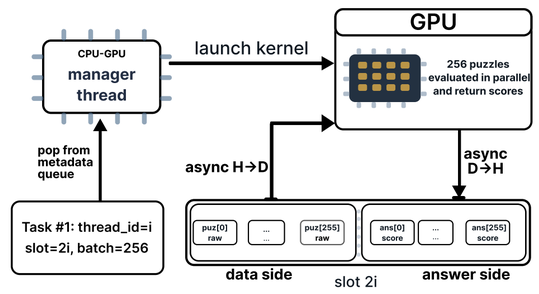
CPU Workers Build and Queue Batches ②

Dense puzzle → Task → slot 2i/2i+1 → metadata-only queue



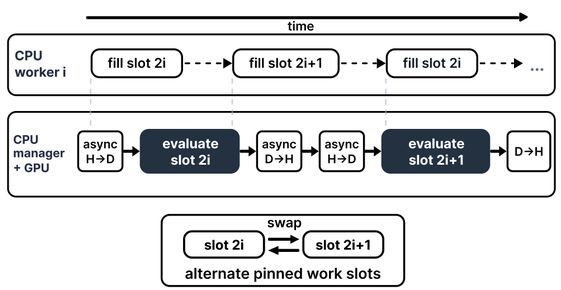
GPU Manager Dispatches Batched Evaluations ③

metadata → slot 2i → async H2D → kernels → async D2H answers

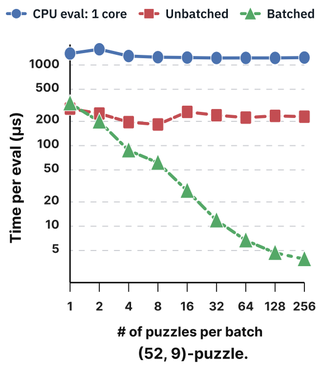


CPU–GPU Overlap: Two-Slot Illustration ④

GPU evaluates slot 2i while worker i fills slot 2i+1; then slots alternate



Batching Amortizes Evaluation Cost



Profiled cost per puzzle (µs)

(52,9) · 1 GPU

Stage	Unbatched	Batched
CPU alloc/free	3351.00	0
GPU kernel	111.56	3.421
GPU H2D+D2H	1.840	0.0203

Sample counts (≈2.8 s)

Kernel launches	212,784	3,094
Puzzles evaluated	17,732	751,760

New discovery: (79, 10) SUSP

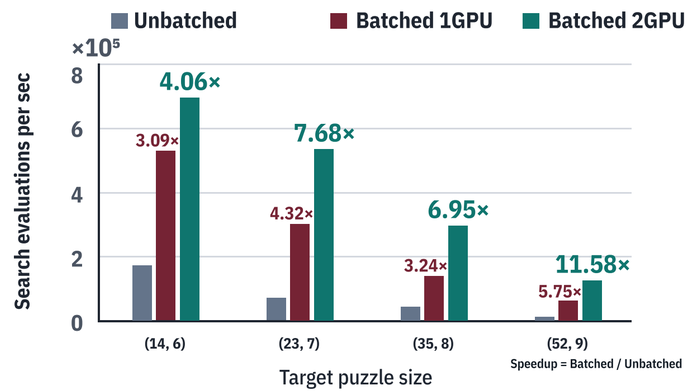
Previous best: (78,10) [2]
New result: (79,10)

We found a **(79, 10) SUSP**, improving on the largest width-10 SUSP previously reported by Anderson and Le [2].

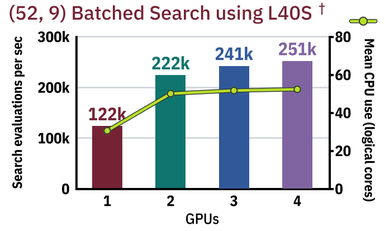
We started searching with Anderson and Le's (78, 10)-SUSP as seed and used the configuration described in the Experiment Setup section.

GitHub repository
matrix-multiplication-search

Batched Search Raises End-to-End Throughput By Up to 11.58x



(52, 9) Batched Search using L40S †



Multi-GPU extension: On host with 80 OpenMP threads, throughput rises 2.05× from 1 to 4 L40S GPUs. The CPU manager thread selects GPUs in **round-robin** order and launches kernels on each device. However, the gains diminish beyond 2 GPUs, motivating further work on resource utilization.

Related Work

CPU–GPU heterogeneous systems have been applied to combinatorial search in real-world applications, including route planning, circuit design, drug discovery, etc. Uni-Dock [3] batches molecular docking on GPUs. We apply batching to SUSP search, where CPU workers expand a changing frontier and feed GPU evaluation through reusable buffers and a metadata-only queue. Our contribution is this search integration and its application to finding a (79,10)-SUSP.

Conclusions & Future Work

We developed a heterogeneous search framework with batched evaluation, multiple CPU frontier-expansion workers, and a dedicated GPU manager to keep the CPU and GPU busy.

Future work will scale the search across distributed multi-node CPU–GPU systems and develop resource-aware scheduling for efficient scaling.

References & Acknowledgements

[1] Cohn et al., "Group-theoretic algorithms for matrix multiplication," FOCS 2005.
[2] Anderson & Le, "Efficiently-verifiable SUSP and matrix multiplication," COCOON 2023.
[3] Y. Yu, C. Cai, J. Wang, Z. Bo, Z. Zhu, and H. Zheng, "Uni-Dock: GPU-accelerated docking enables ultralarge virtual screening," Journal of Chemical Theory and Computation, vol. 19, no. 11, pp. 3336–3345, 2023
† CPU and GPU resources for the L40S runs were rented from Modal.
Acknowledgments: This work was supported by Union College's Summer Research program.

Experimental Environment

- **CPU:** AMD Ryzen 9 9900X 12 cores/ 24 threads
- **GPUs:** NVIDIA GeForce RTX 5070 Ti 16 GiB x2
- **RAM:** 64 GiB
- Unbatched: 24 CPU threads
- GPU batch: 256/task
- Throughput: mean of 10 runs
- All tests use this environment unless further described

AI-Use Disclosure
Generative AI tools were used for language editing in (*) sentences. Technical content, measurements, interpretations, and citations were independently verified. OpenAI Codex (GPT-5), accessed July 2026.