THE UNIVERSITY of ADELAIDE

APPROXIMATE APPROACHES TO THE TRAVELING THIEF PROBLEM

Faulkner, Polyakovskiy, Schultz, Wagner

Hayden Faulkner
GECCO 2015

seek LIGHT

OUTLINE

- Address the multifaceted Traveling Thief Problem (TTP)
- Introduce a new fast basic heuristic method for achieving a good packing of items provided a tour
- Introduce two additional operators, one of which alters a packed tour based on packing
- Compare varying combinations and setups of the heuristic and operators

THE TRAVELING THIEF PROBLEM (TTP)

- n cities, with distances $d(i,j)$ between cities i and j
- m items, each with weight w_{ik} and profit p_{ik}
- Knapsack capacity W
- Renting rate R
- v_{min} and v_{max} representing the minimal and maximal speed of the traveller

THE TRAVELING THIEF PROBLEM (TTP)

- <u>Goal:</u> Visit each city exactly once, maximising the total profit P such that the total weight does not exceed the knapsack capacity W, where P is defined as:

$$P = \sum_{i=1}^{m} p_i \, x_i \; - R \sum_{i=1}^{n} t_{i,i+1}$$

where $x_i = \{1|0\}$ depending on whether the item i is picked $\{1\}$ or not $\{0\}$, and $t_{i,j}$ is defined as:

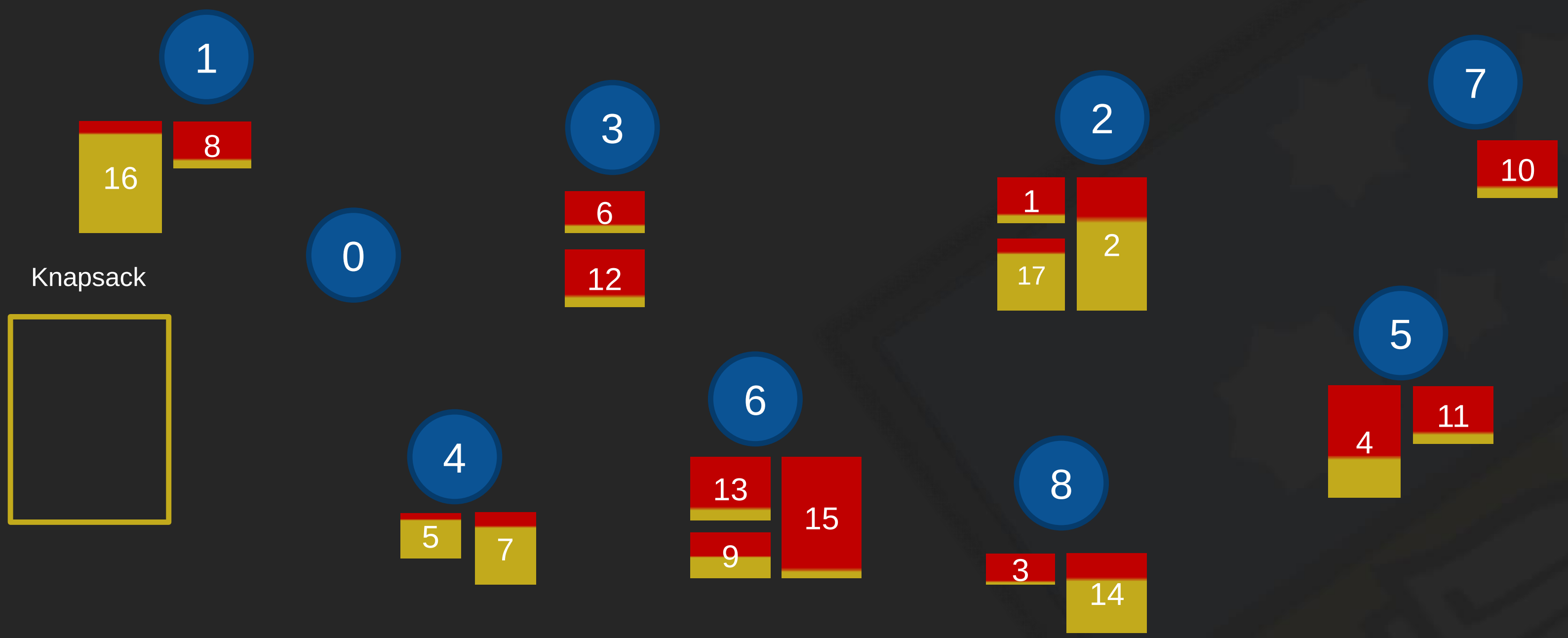
$$t_{i,j} = \frac{d(\Pi_i, \Pi_j)}{v_{max} - W_{\Pi_i}\left(\frac{v_{max} - v_{min}}{W}\right)}$$

where Π_i is the city at tour position i in tour the Π, and W_{Π_i} is the current weight of the knapsack at city Π_i.

THE UNIVERSITY of ADELAIDE

THE TRAVELING THIEF PROBLEM (TTP)

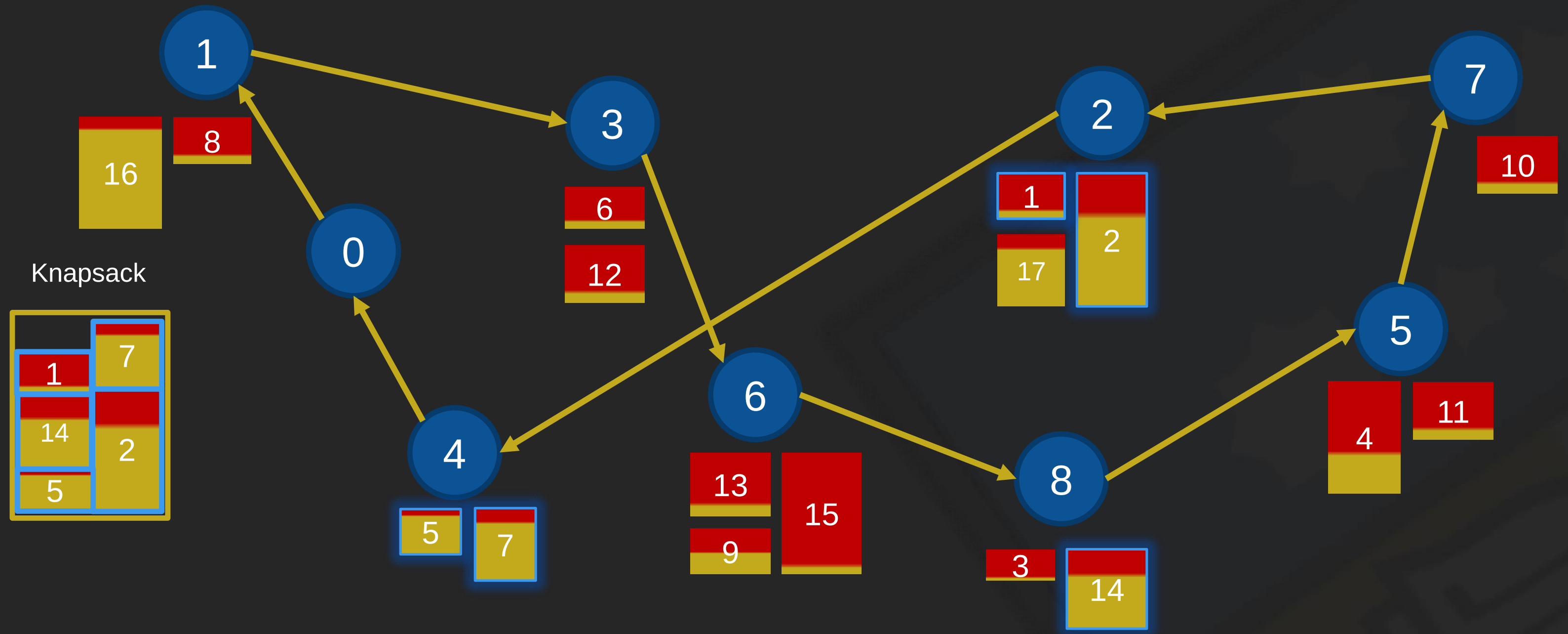
- Composed of the merging of the Traveling Salesman Problem and the Knapsack Problem



THE UNIVERSITY of ADELAIDE

THE TRAVELING THIEF PROBLEM (TTP)

- Composed of the merging of the Traveling Salesman Problem and the Knapsack Problem



THE UNIVERSITY of ADELAIDE

FAST PACKING ROUTINE USING BASIC HEURISTIC

- Finds a TSP solution using Chained-Lin-Kernighan [1]
- Using the fixed TSP solution, generates a solution for KP problem
- Ignores the interdependency between the individual TSP and KP problems

FAST PACKING ROUTINE USING BASIC HEURISTIC

FAST PACKING ROUTINE USING BASIC HEURISTIC

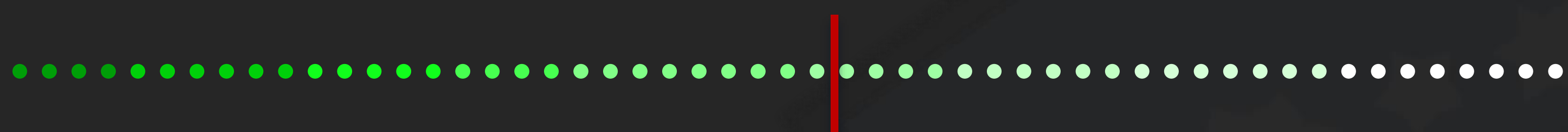
- Calculate heuristic score s_{ik} for each item

FAST PACKING ROUTINE USING BASIC HEURISTIC

- Calculate heuristic score s_{ik} for each item
- Sorts items in non-decreasing order based on score s_{ik}

FAST PACKING ROUTINE USING BASIC HEURISTIC

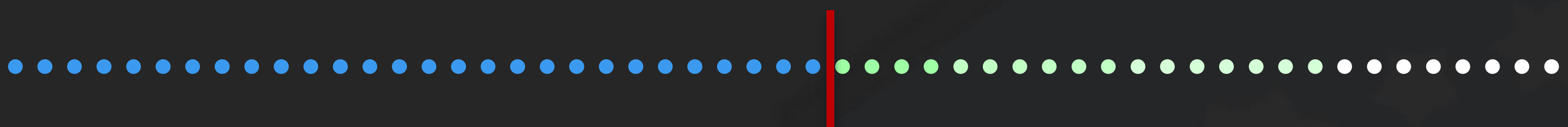
- Calculate heuristic score s_{ik} for each item
- Sorts items in non-decreasing order based on score s_{ik}
- Greedily adds items to the packing plan until objective value no longer increases



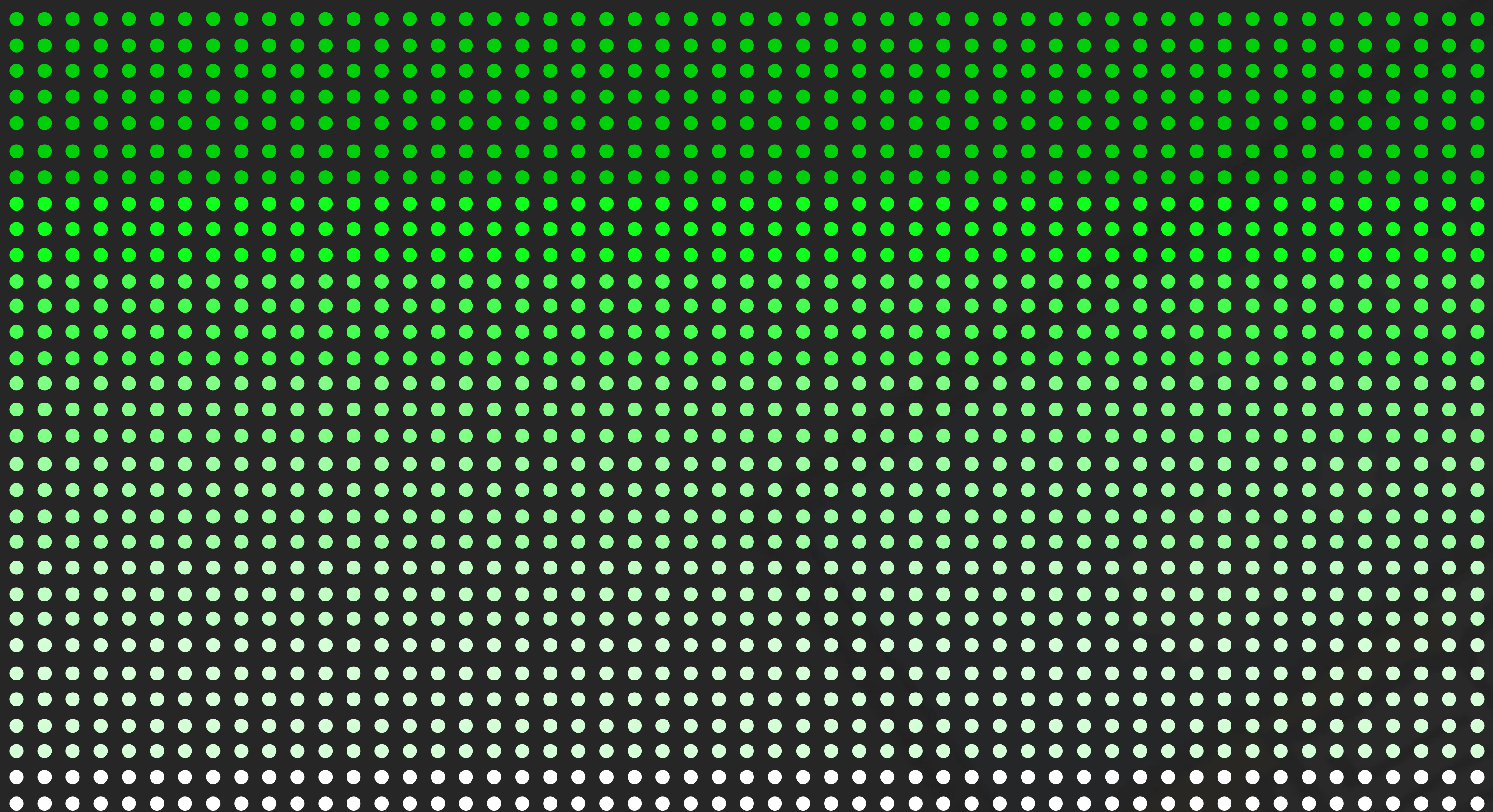
THE UNIVERSITY *of* ADELAIDE

FAST PACKING ROUTINE USING BASIC HEURISTIC

- Calculate heuristic score s_{ik} for each item
- Sorts items in non-decreasing order based on score s_{ik}
- Greedily adds items to the packing plan until objective value no longer increases

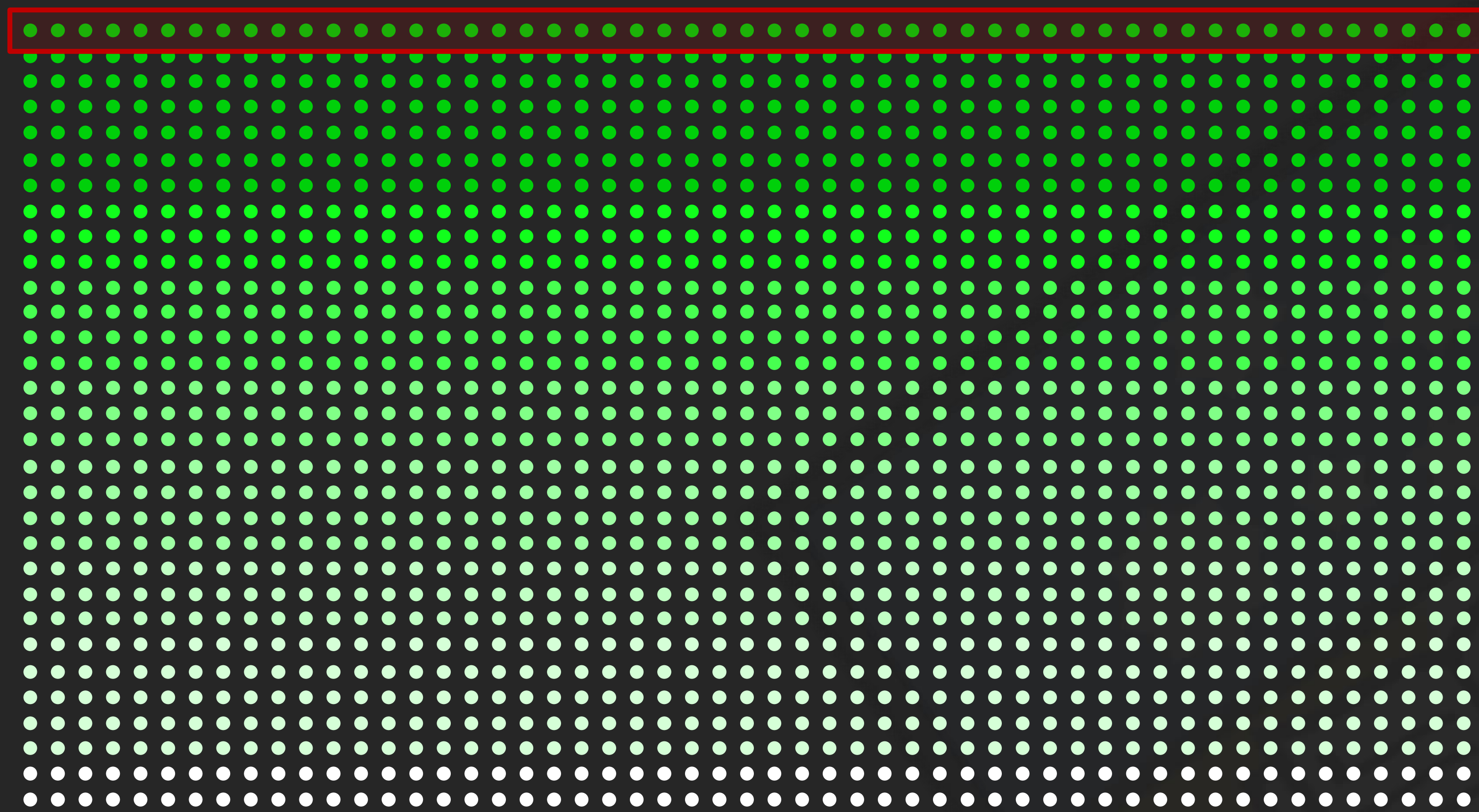


FAST PACKING ROUTINE USING BASIC HEURISTIC

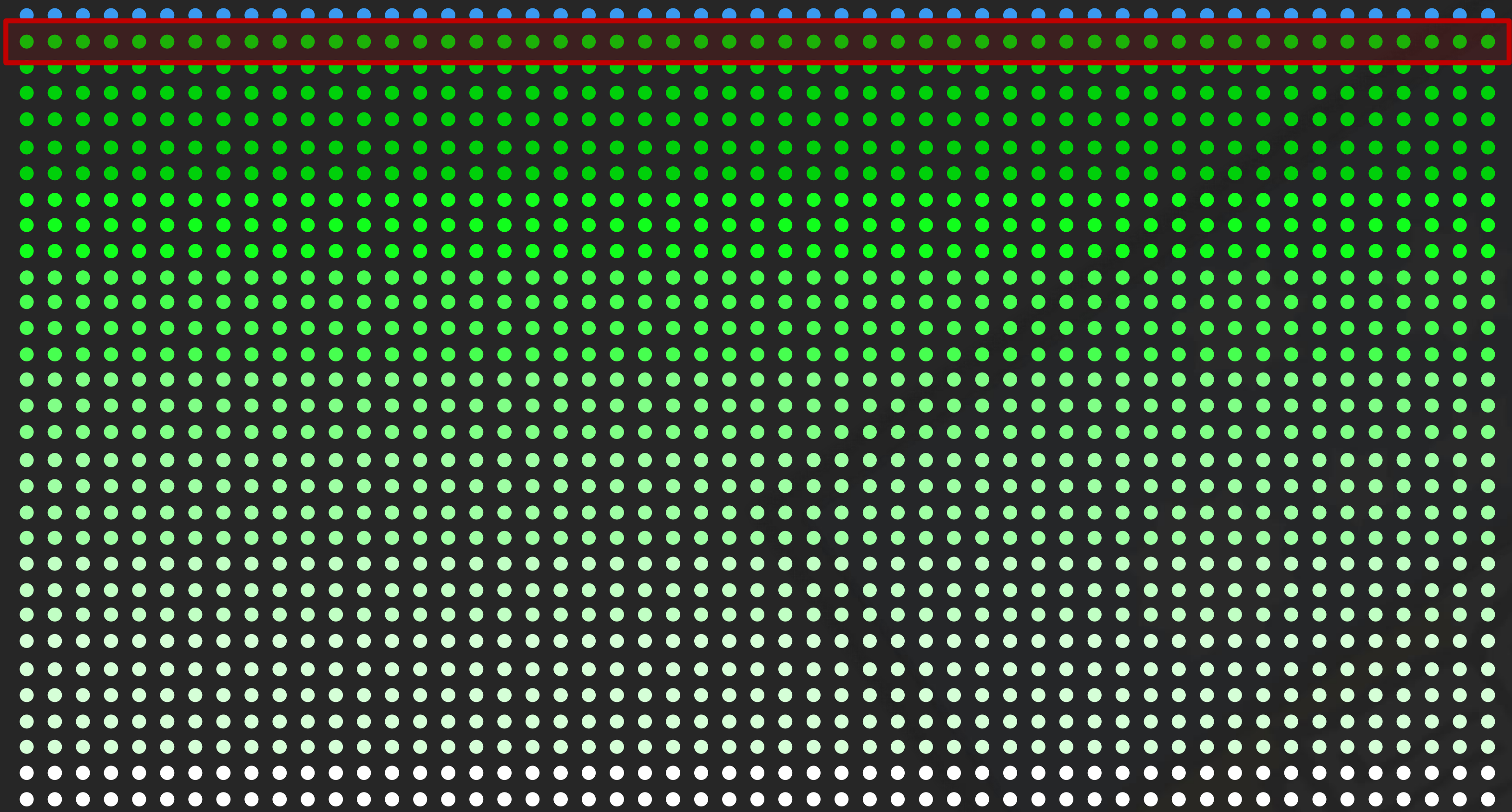


FAST PACKING ROUTINE USING BASIC HEURISTIC

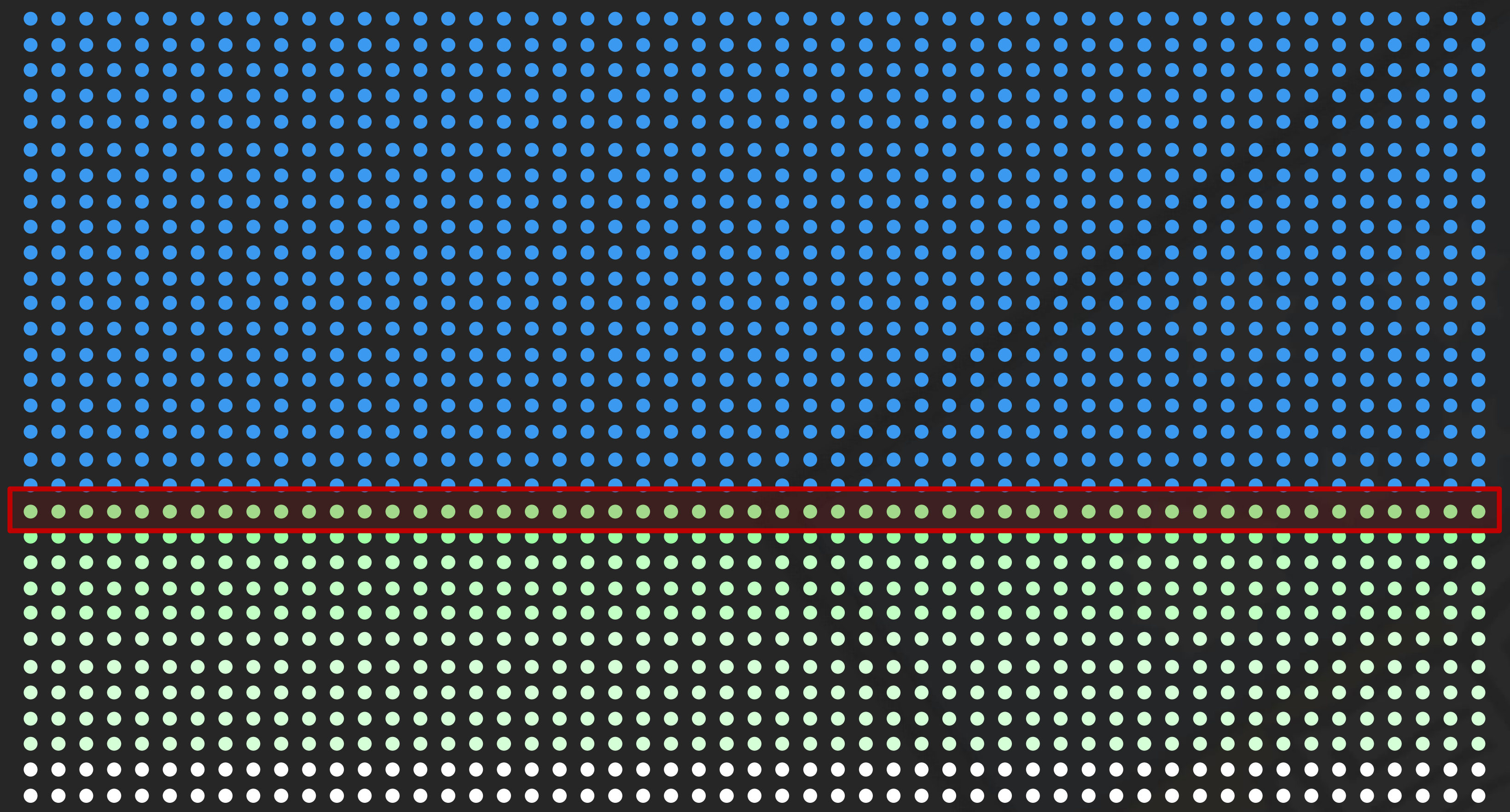
$$v = \lfloor {}^{m}/_{\tau} \rfloor$$



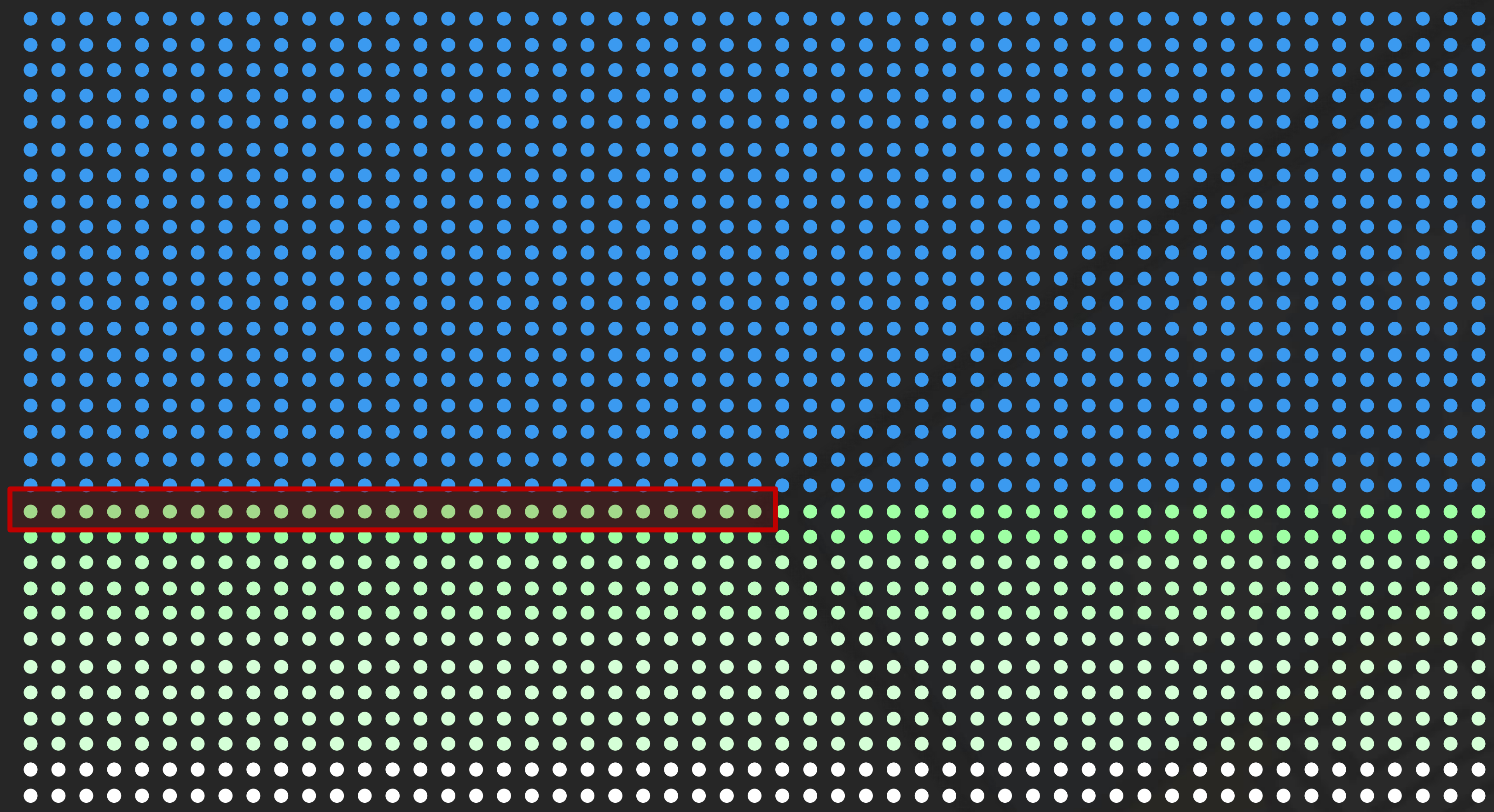
FAST PACKING ROUTINE USING BASIC HEURISTIC



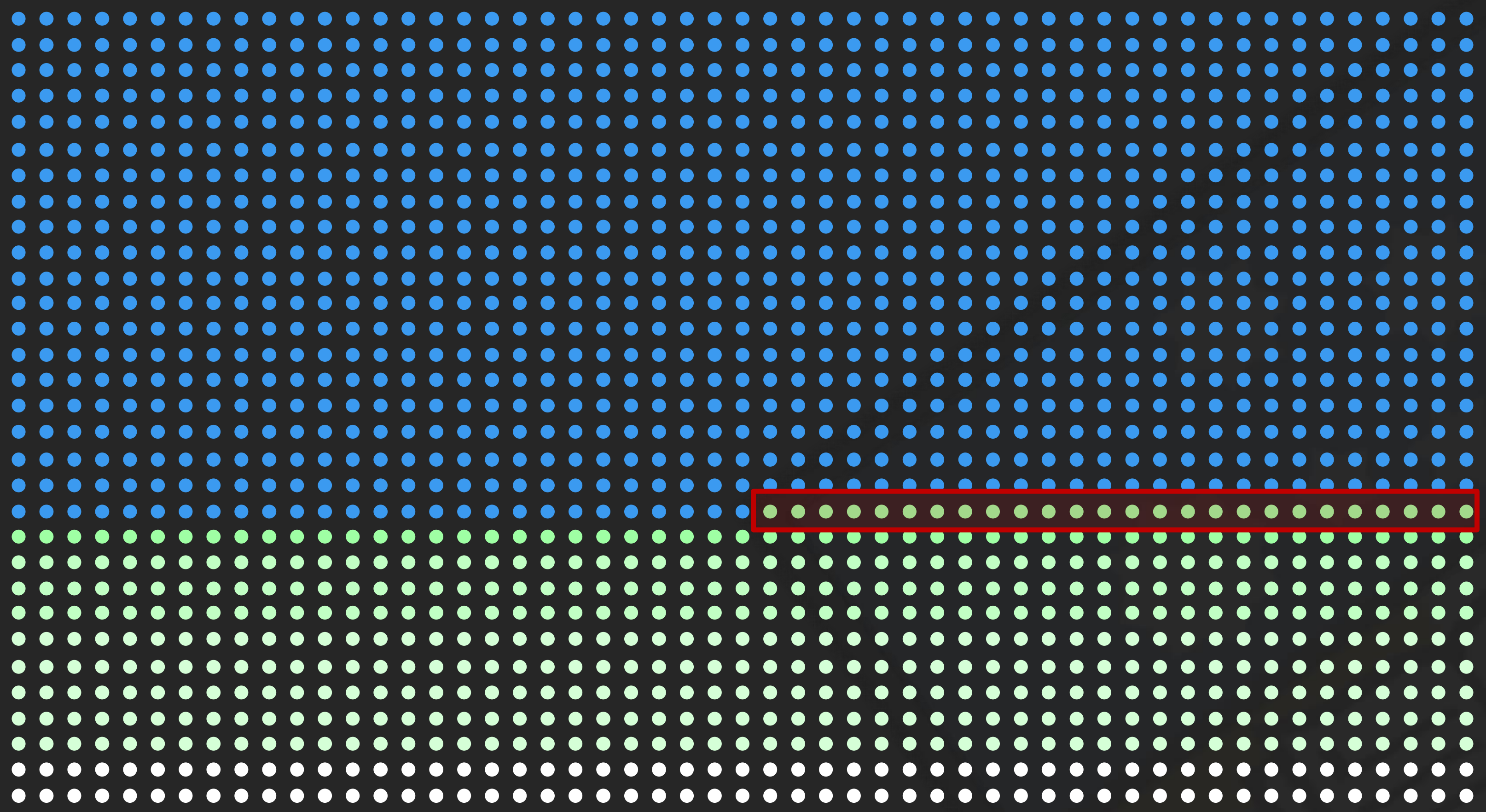
FAST PACKING ROUTINE USING BASIC HEURISTIC



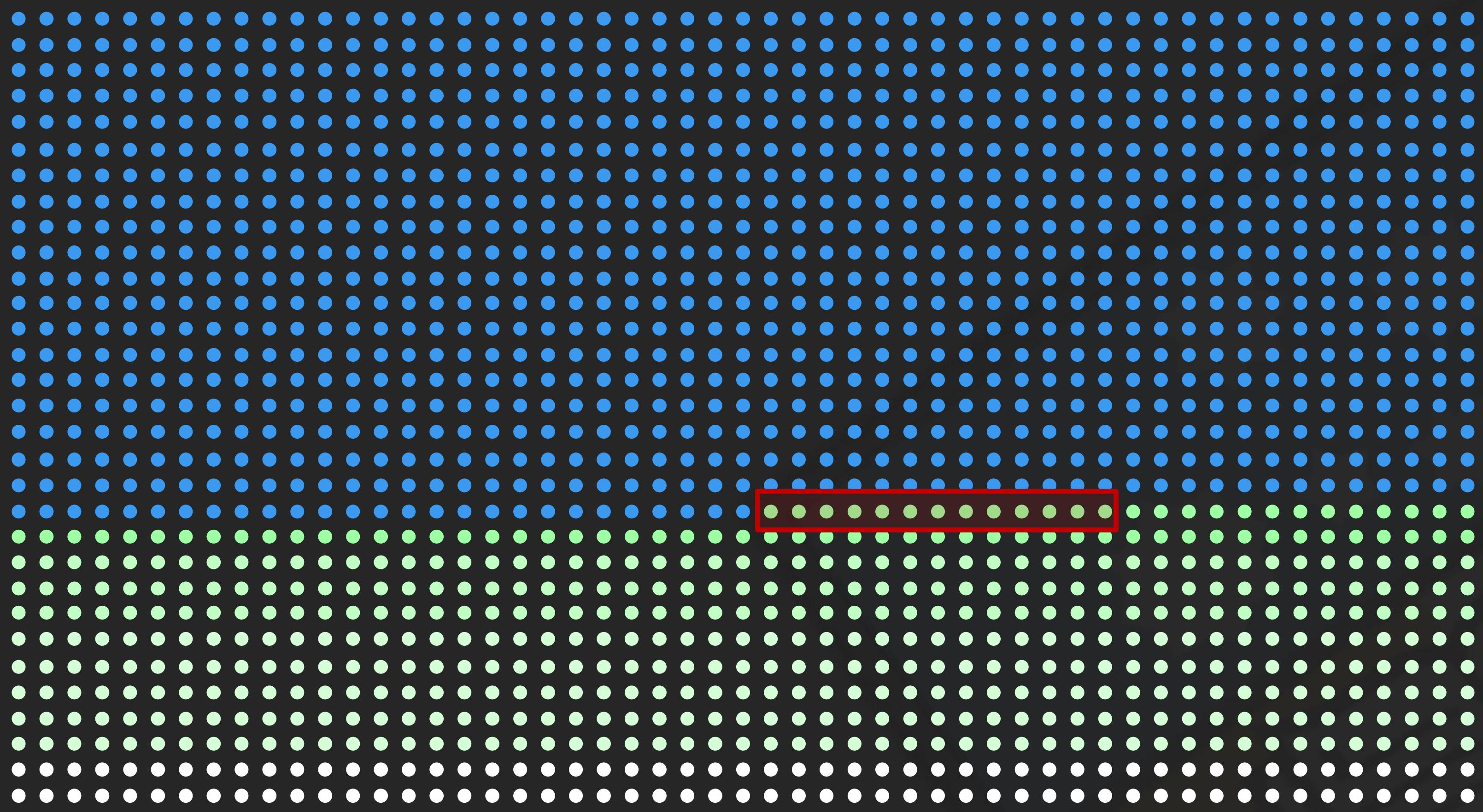
FAST PACKING ROUTINE USING BASIC HEURISTIC



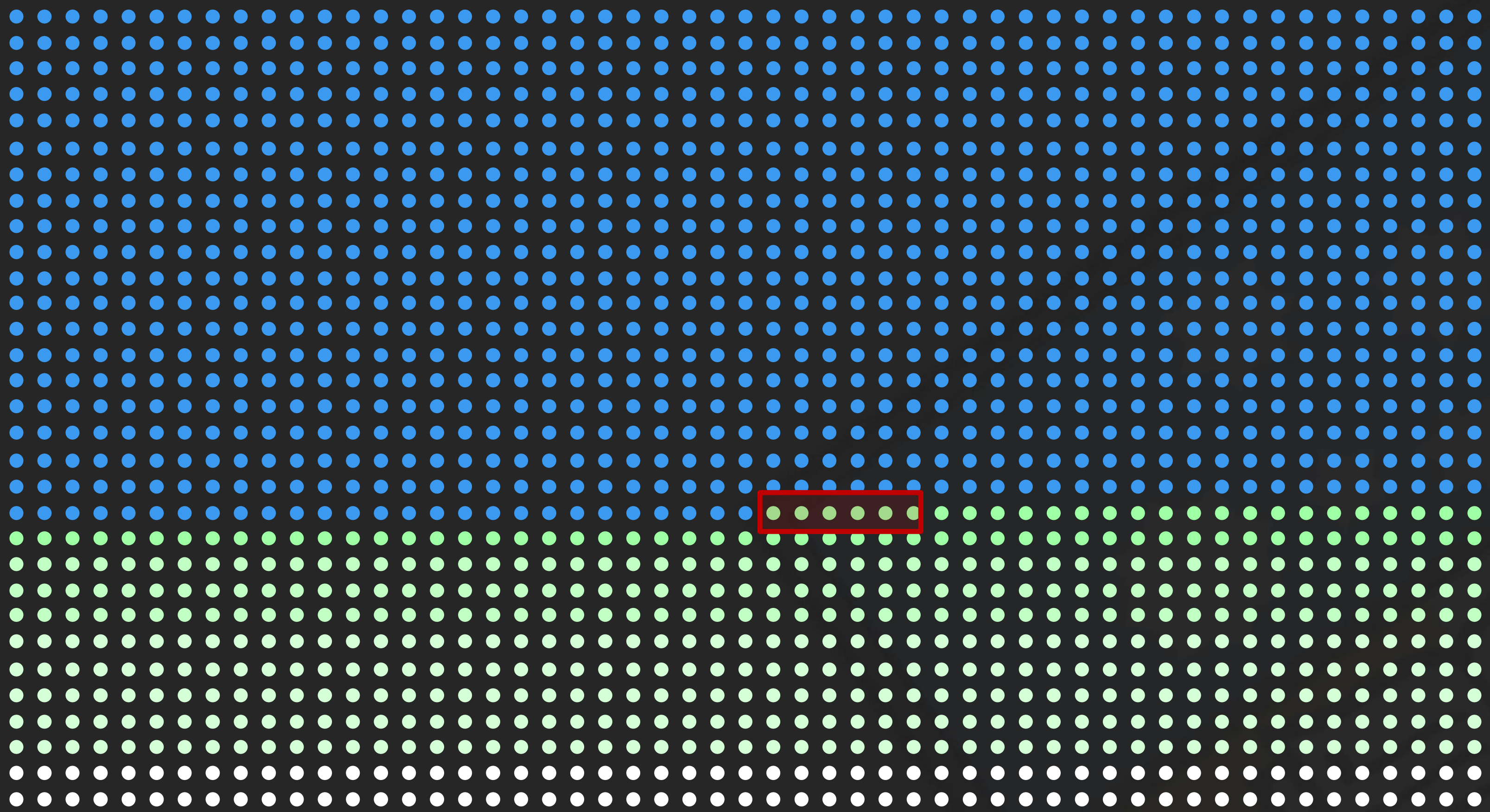
FAST PACKING ROUTINE USING BASIC HEURISTIC



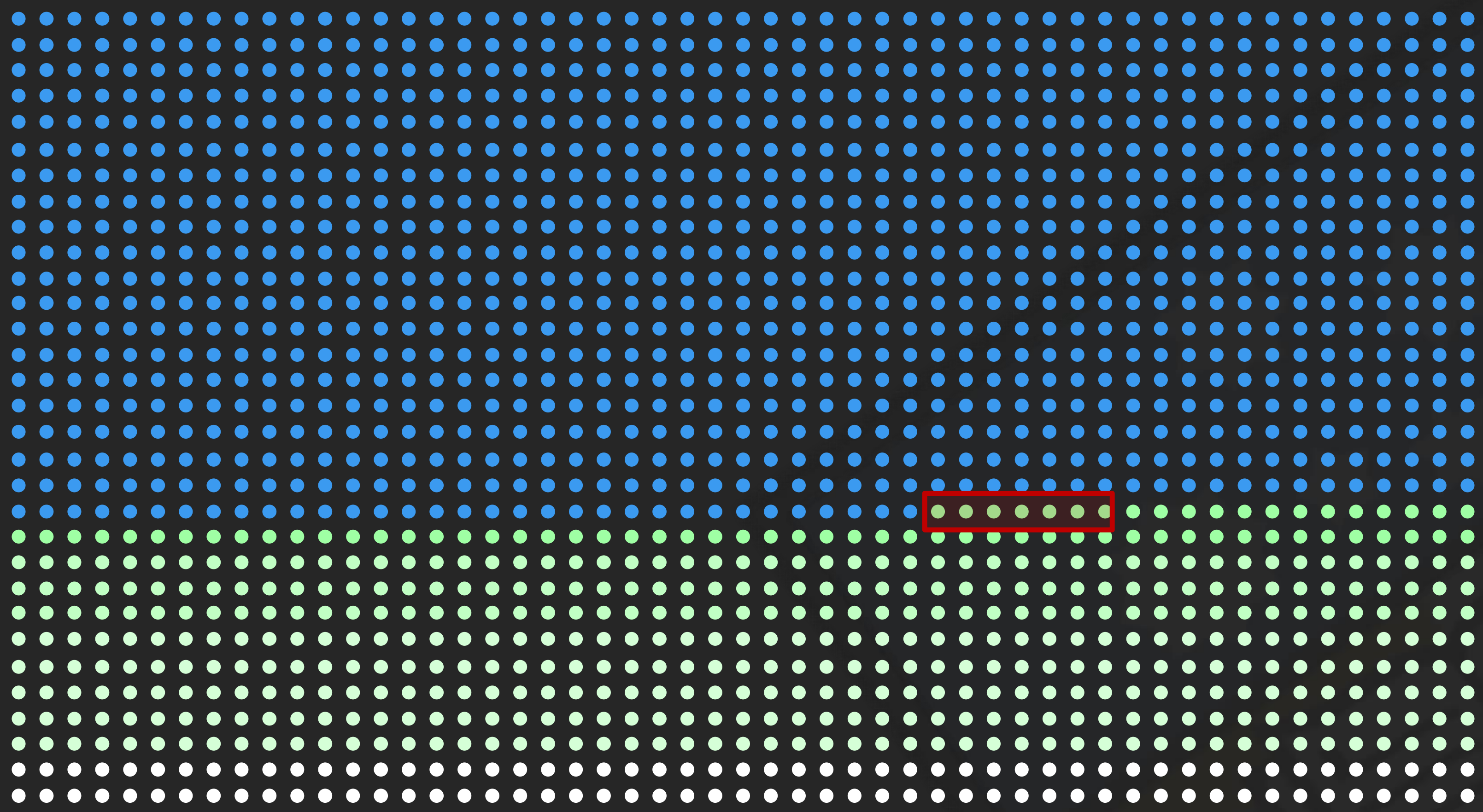
FAST PACKING ROUTINE USING BASIC HEURISTIC



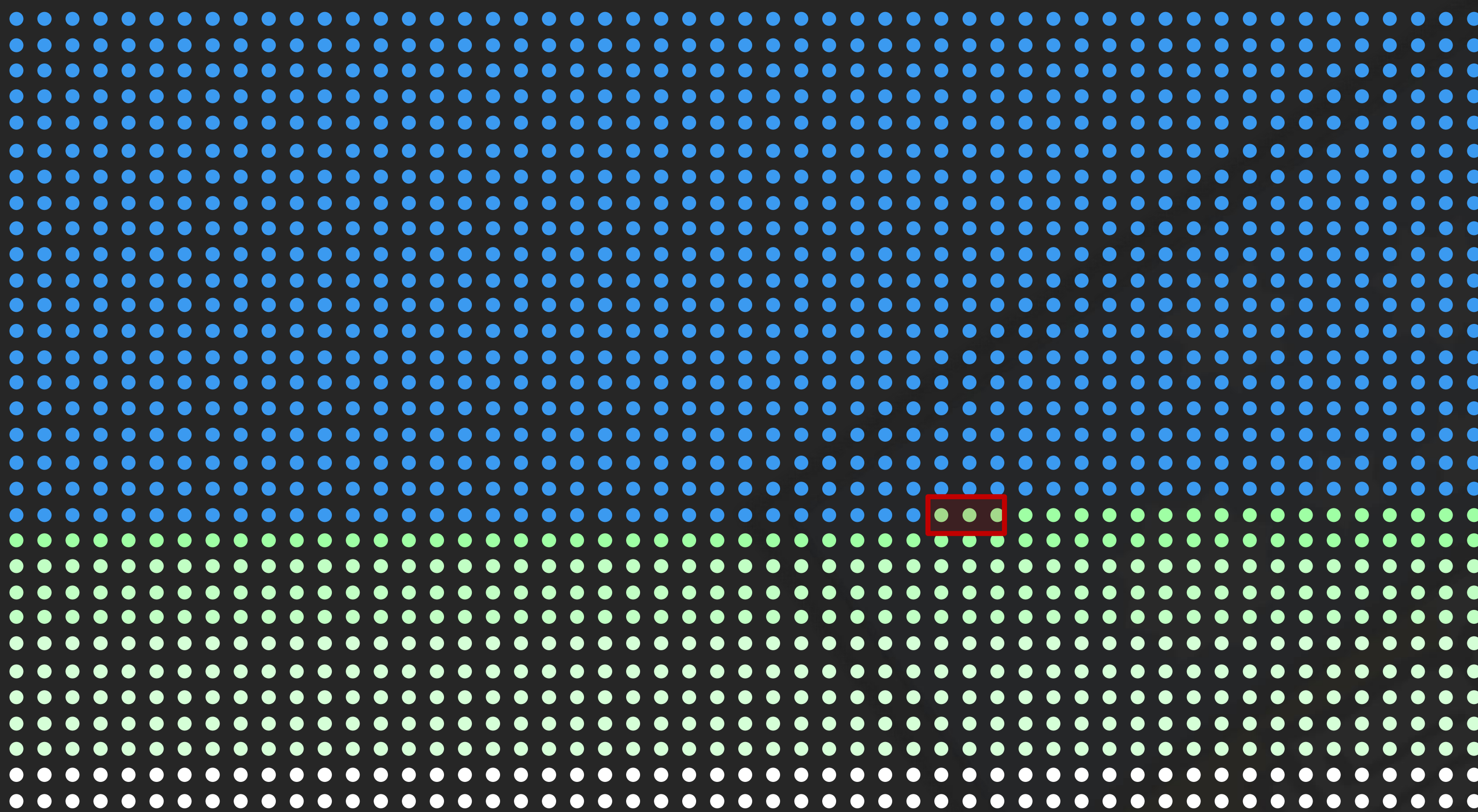
FAST PACKING ROUTINE USING BASIC HEURISTIC



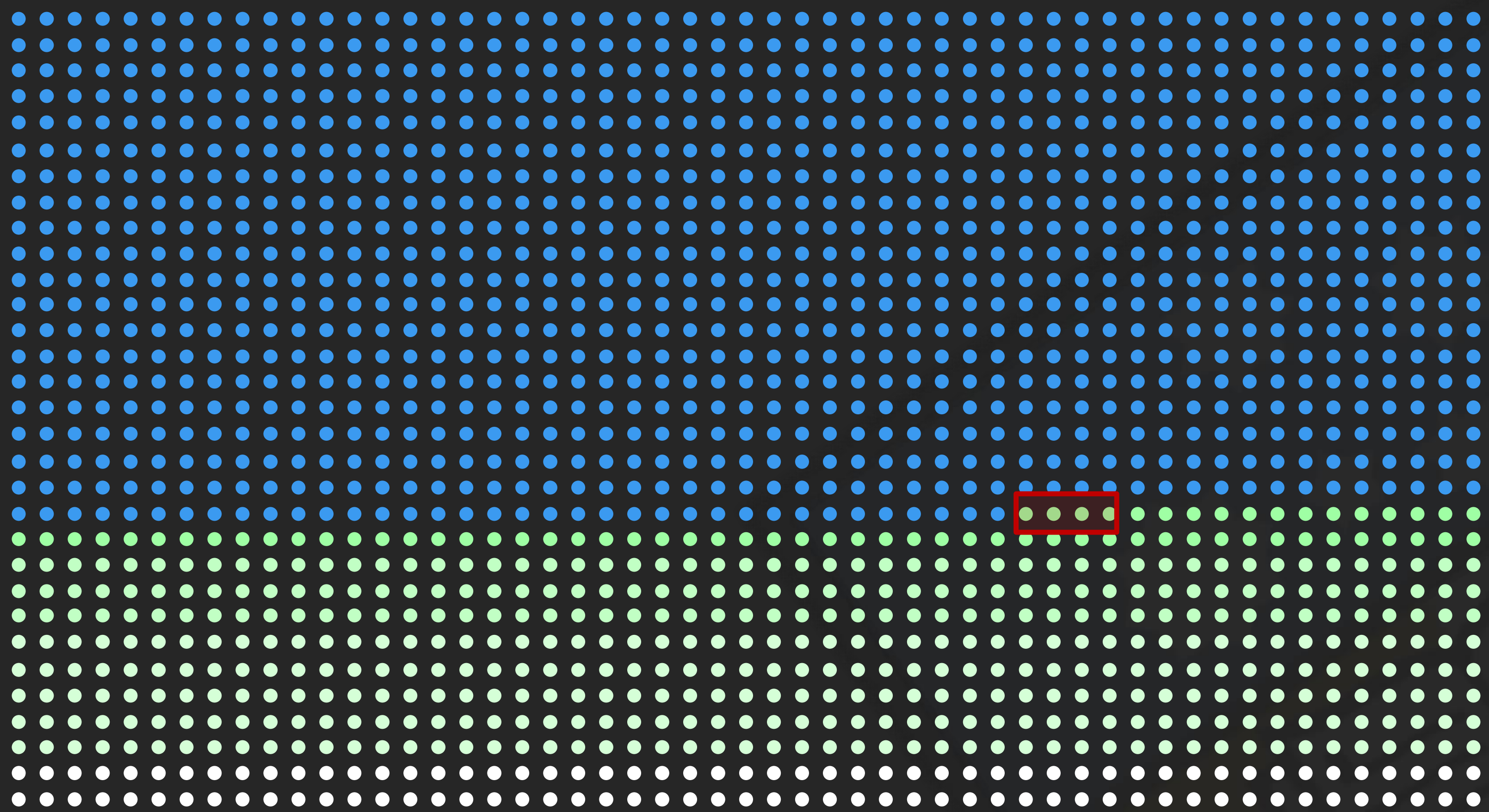
FAST PACKING ROUTINE USING BASIC HEURISTIC



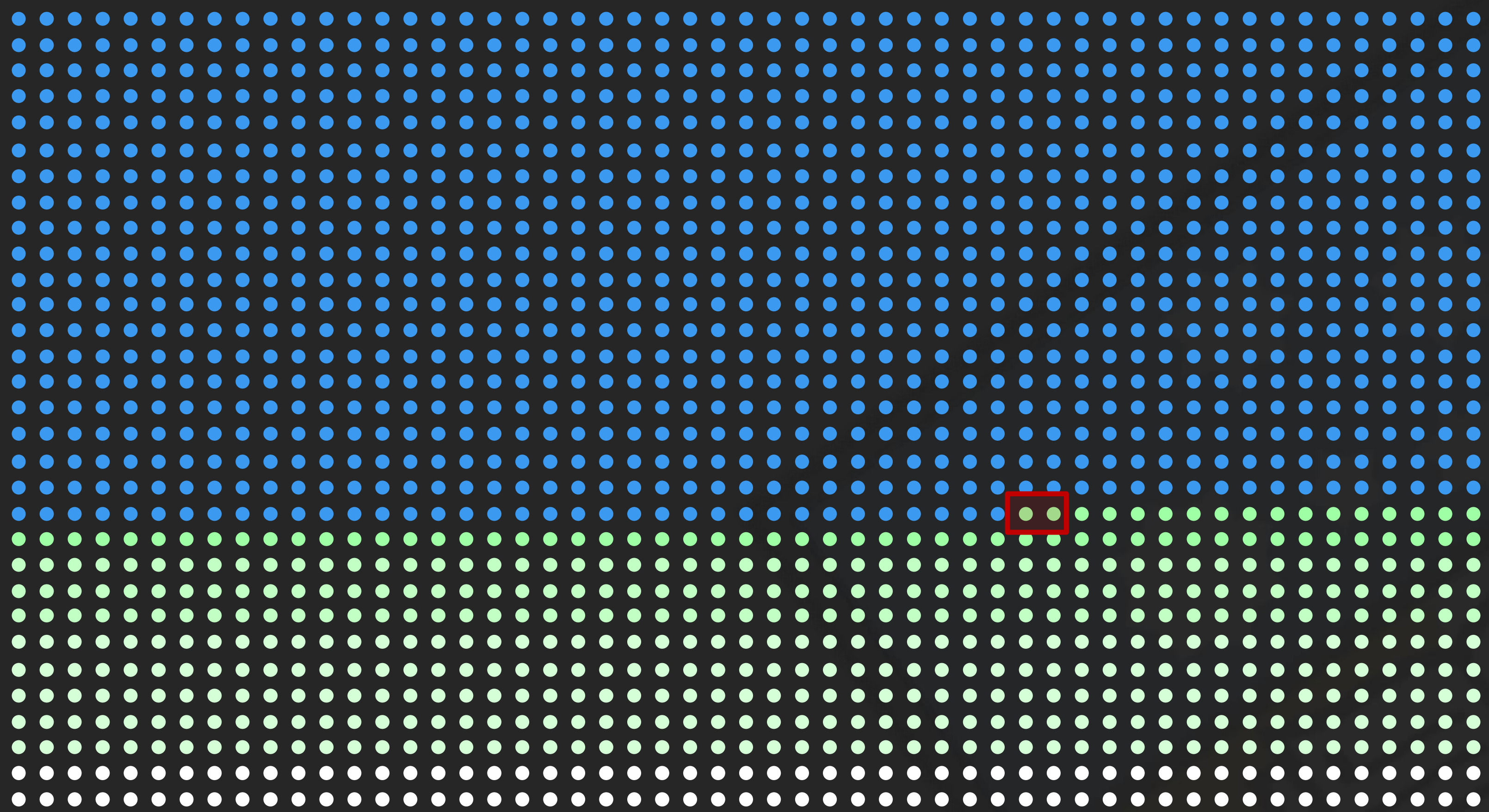
FAST PACKING ROUTINE USING BASIC HEURISTIC



FAST PACKING ROUTINE USING BASIC HEURISTIC

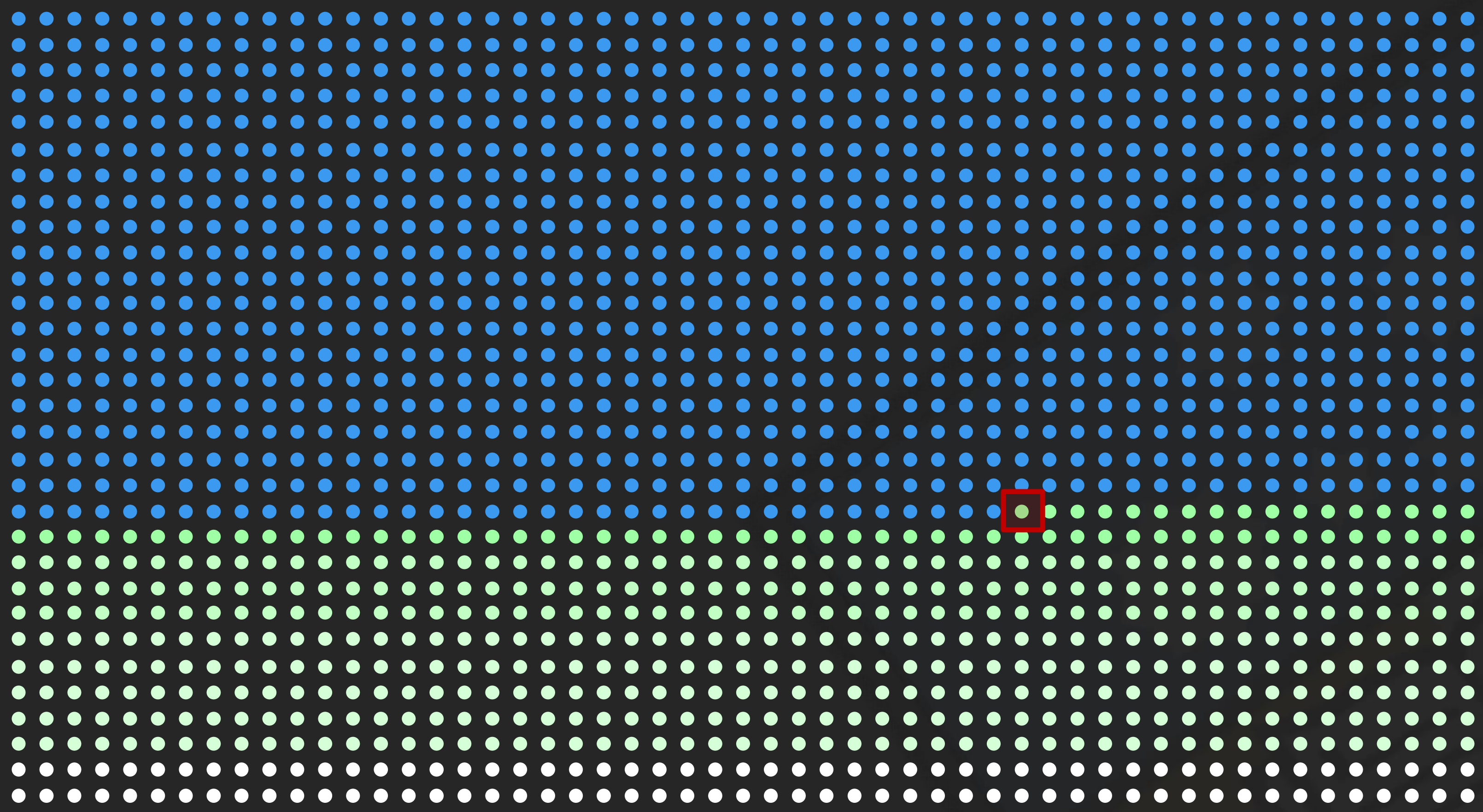


FAST PACKING ROUTINE USING BASIC HEURISTIC

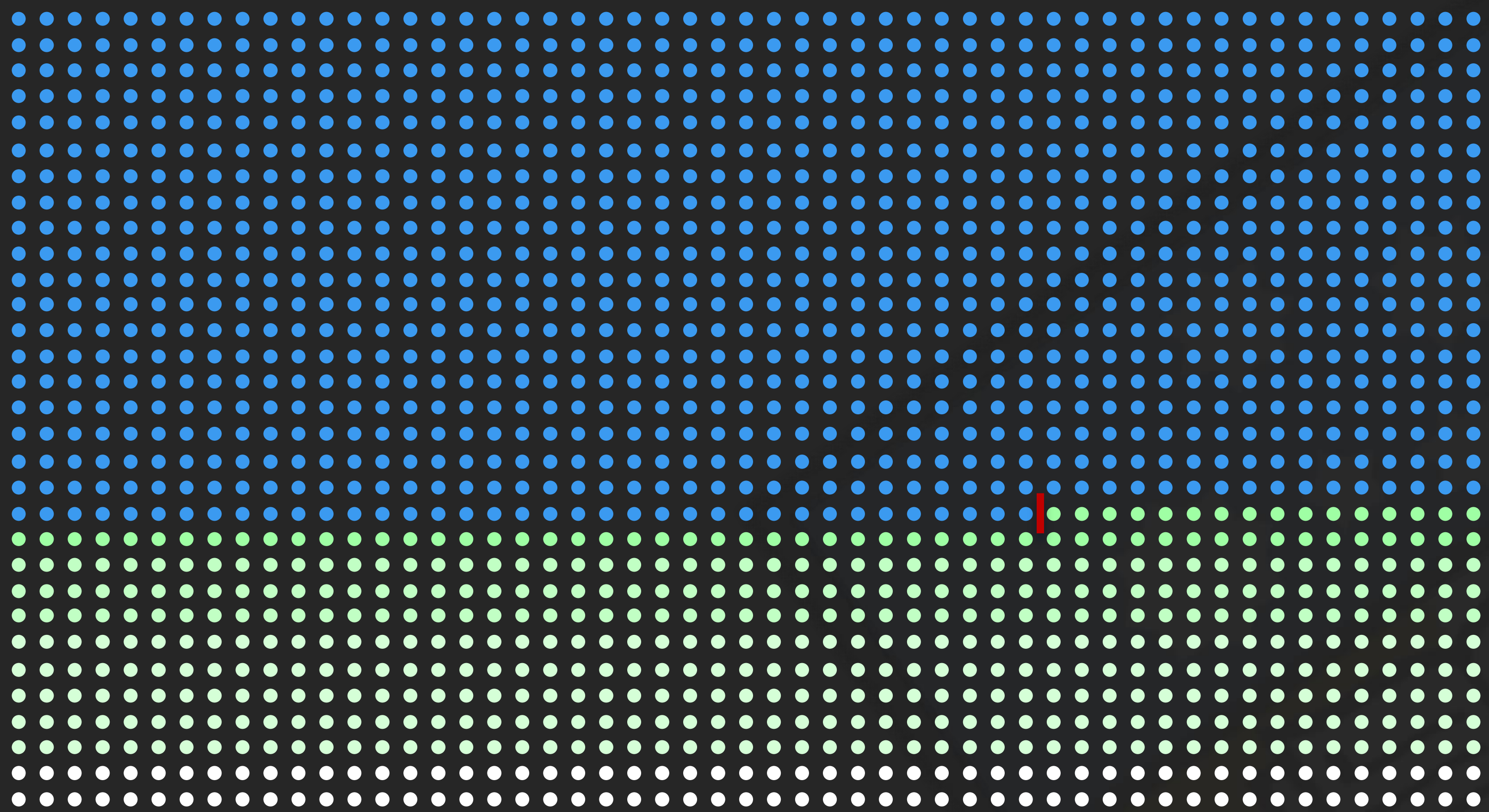


THE UNIVERSITY of ADELAIDE

FAST PACKING ROUTINE USING BASIC HEURISTIC



FAST PACKING ROUTINE USING BASIC HEURISTIC



FAST BASIC HEURISTIC

- Once a fixed tour calculated, a score s_{ik} is calculated for each item k in city i:

$$s_{ik} = \frac{p_{ik}}{w_{ik}}$$

where p_{ik} and w_{ik} is the profit and weight of item k respectively.

FAST BASIC HEURISTIC

- Once a fixed tour calculated, a score s_{ik} is calculated for each item k in city i:

$$s_{ik} = \frac{p_{ik}}{w_{ik} \times d_i}$$

where p_{ik} and w_{ik} is the profit and weight of item k respectively, and d_i is the distance from city i to the end of the tour.

THE UNIVERSITY of ADELAIDE

FAST BASIC HEURISTIC

- Once a fixed tour calculated, a score s_{ik} is calculated for each item k in city i:

$$s_{ik} = \frac{p_{ik}^{x}}{w_{ik}^{y} \times d_i^{z}}$$

where p_{ik} and w_{ik} is the profit and weight of item k respectively, and d_i is the distance from city i to the end of the tour.

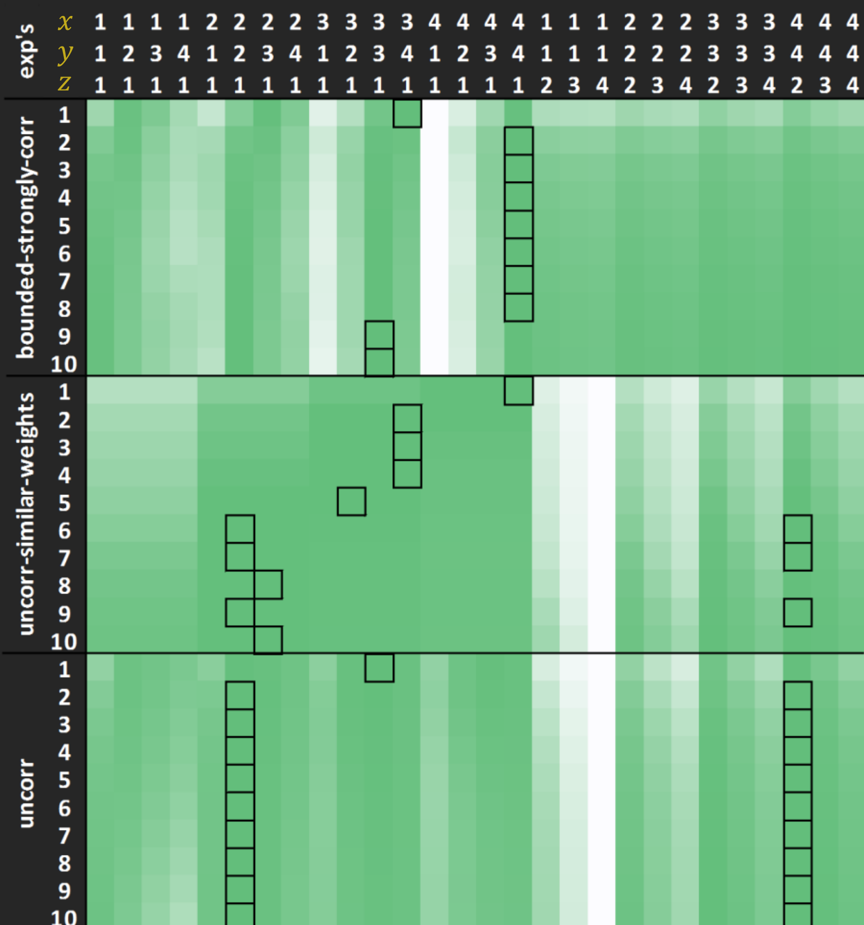
THE UNIVERSITY of ADELAIDE

FAST BASIC HEURISTIC

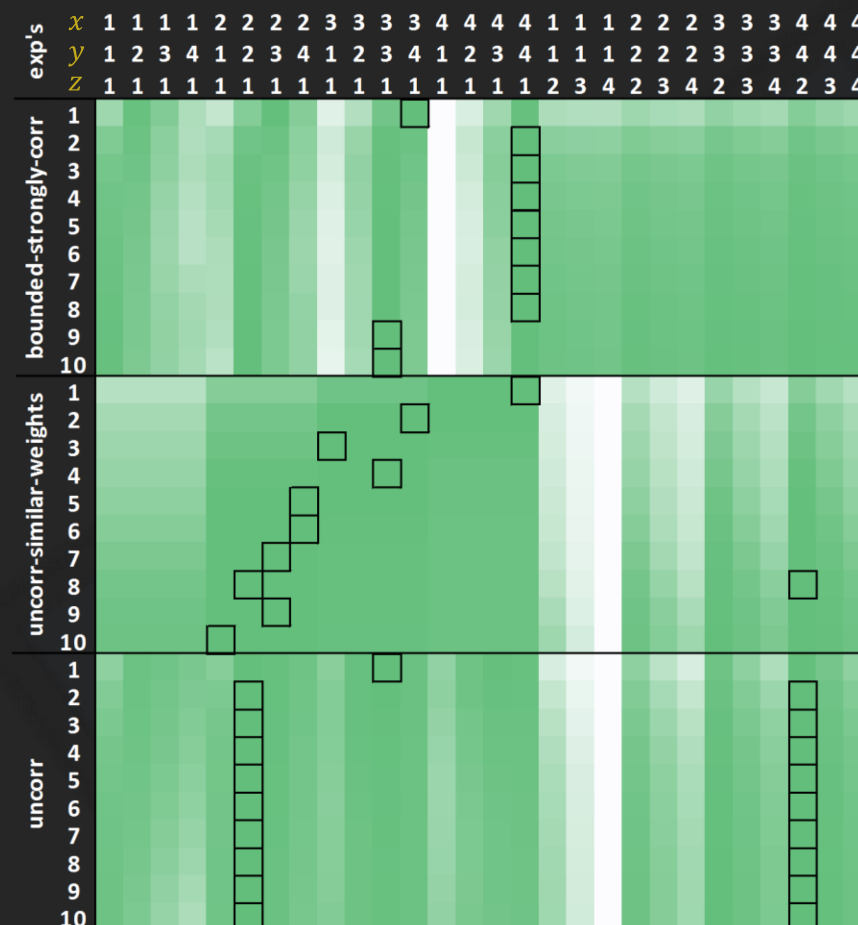
exp's	x	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4	1	1	1	2	2	2	3	3	3	4	4	4
	y	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	1	1	2	2	2	3	3	3	4	4	4
	z	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	3	4	2	3	4	2	3	4	2	3	4
bounded-strongly-corr	1																												
	2																												
	3																												
	4																												
	5																												
	6																												
	7																												
	8																												
	9																												
	10																												
uncorr-similar-weights	1																												
	2																												
	3																												
	4																												
	5																												
	6																												
	7																												
	8																												
	9																												
	10																												
uncorr	1																												
	2																												
	3																												
	4																												
	5																												
	6																												
	7																												
	8																												
	9																												
	10																												

exp's	x	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4	1	1	1	2	2	2	3	3	3	4	4	4
	y	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	1	1	2	2	2	3	3	3	4	4	4
	z	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	3	4	2	3	4	2	3	4	2	3	4
bounded-strongly-corr	1																												
	2																												
	3																												
	4																												
	5																												
	6																												
	7																												
	8																												
	9																												
	10																												
uncorr-similar-weights	1																												
	2																												
	3																												
	4																												
	5																												
	6																												
	7																												
	8																												
	9																												
	10																												
uncorr	1																												
	2																												
	3																												
	4																												
	5																												
	6																												
	7																												
	8																												
	9																												
	10																												

FAST BASIC HEURISTIC



INSTANCE: rl11849_n118480



INSTANCE: pcb3038_n9111

*objective scores averaged over 50 unique tours

THE UNIVERSITY of ADELAIDE

FAST BASIC HEURISTIC

- Once a fixed tour calculated, a score s_{ik} is calculated for each item k in city i:

$$s_{ik} = \frac{p_{ik}^{\alpha}}{w_{ik}^{\alpha} \times d_i}$$

where p_{ik} and w_{ik} is the profit and weight of item k respectively, and d_i is the distance from city i to the end of the tour.

THE UNIVERSITY of ADELAIDE

FAST BASIC HEURISTIC

HOW TO FIND α?

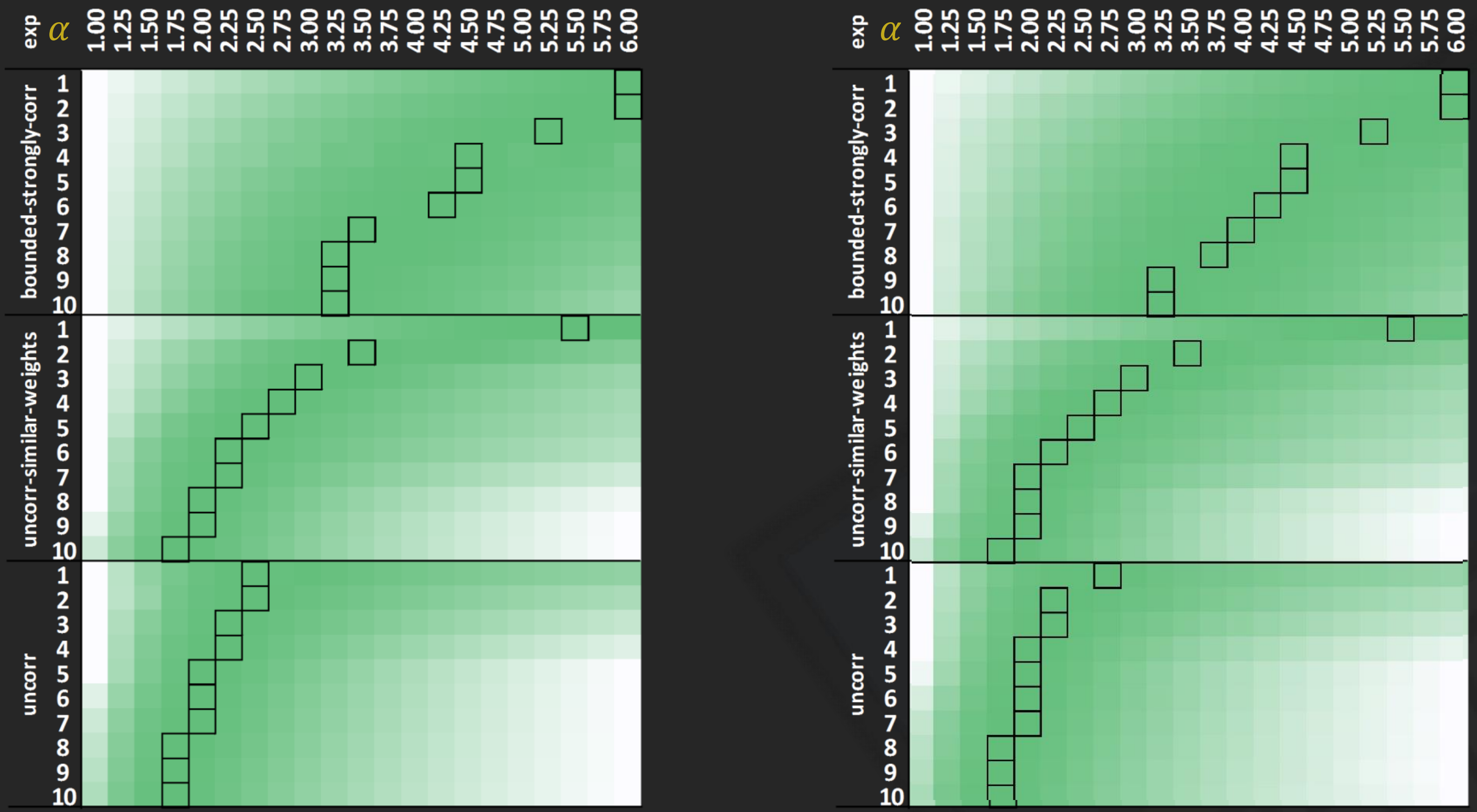
THE UNIVERSITY of ADELAIDE

FAST BASIC HEURISTIC

exp	α	1.00	1.25	1.50	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	3.75	4.00	4.25	4.50	4.75	5.00	5.25	5.50	5.75	6.00
bounded-strongly-corr	1																					
	2																					
	3																					
	4																					
	5																					
	6																					
	7																					
	8																					
	9																					
	10																					
uncorr-similar-weights	1																					
	2																					
	3																					
	4																					
	5																					
	6																					
	7																					
	8																					
	9																					
	10																					
uncorr	1																					
	2																					
	3																					
	4																					
	5																					
	6																					
	7																					
	8																					
	9																					
	10																					

exp	α	1.00	1.25	1.50	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	3.75	4.00	4.25	4.50	4.75	5.00	5.25	5.50	5.75	6.00
bounded-strongly-corr	1																					
	2																					
	3																					
	4																					
	5																					
	6																					
	7																					
	8																					
	9																					
	10																					
uncorr-similar-weights	1																					
	2																					
	3																					
	4																					
	5																					
	6																					
	7																					
	8																					
	9																					
	10																					
uncorr	1																					
	2																					
	3																					
	4																					
	5																					
	6																					
	7																					
	8																					
	9																					
	10																					

FAST BASIC HEURISTIC



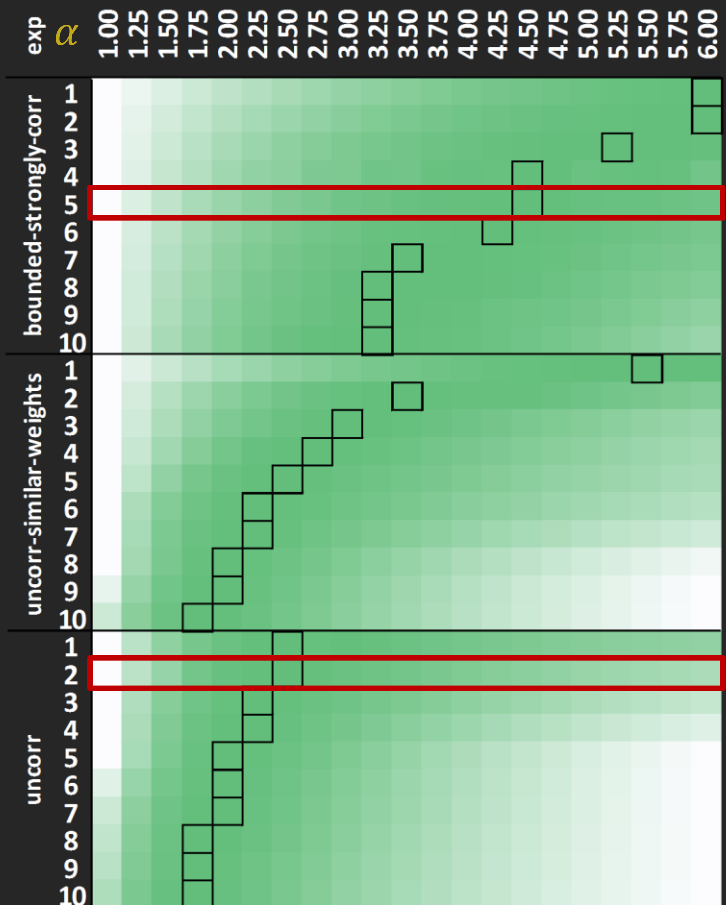
INSTANCE: rl11849_n118480

INSTANCE: pcb3038_n9111

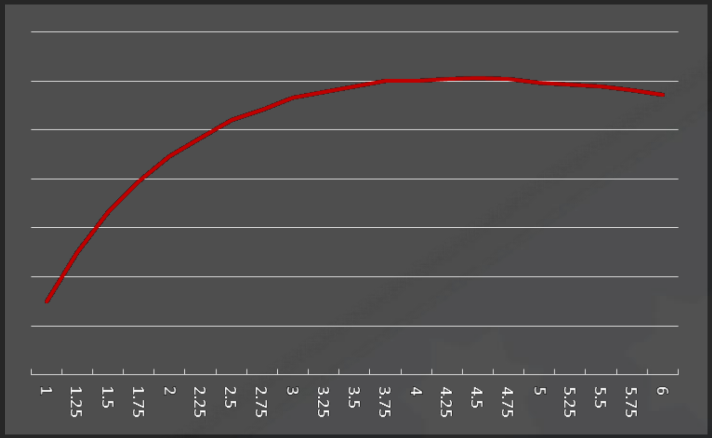
*objective scores averaged over 50 unique tours

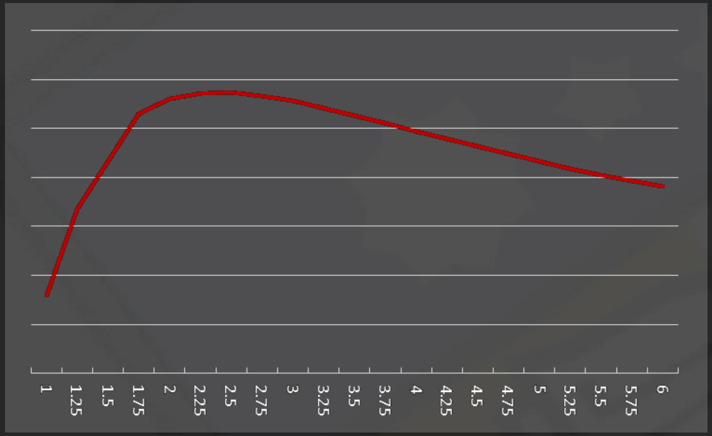
THE UNIVERSITY of ADELAIDE

FAST BASIC HEURISTIC



INSTANCE: rl11849_n118480



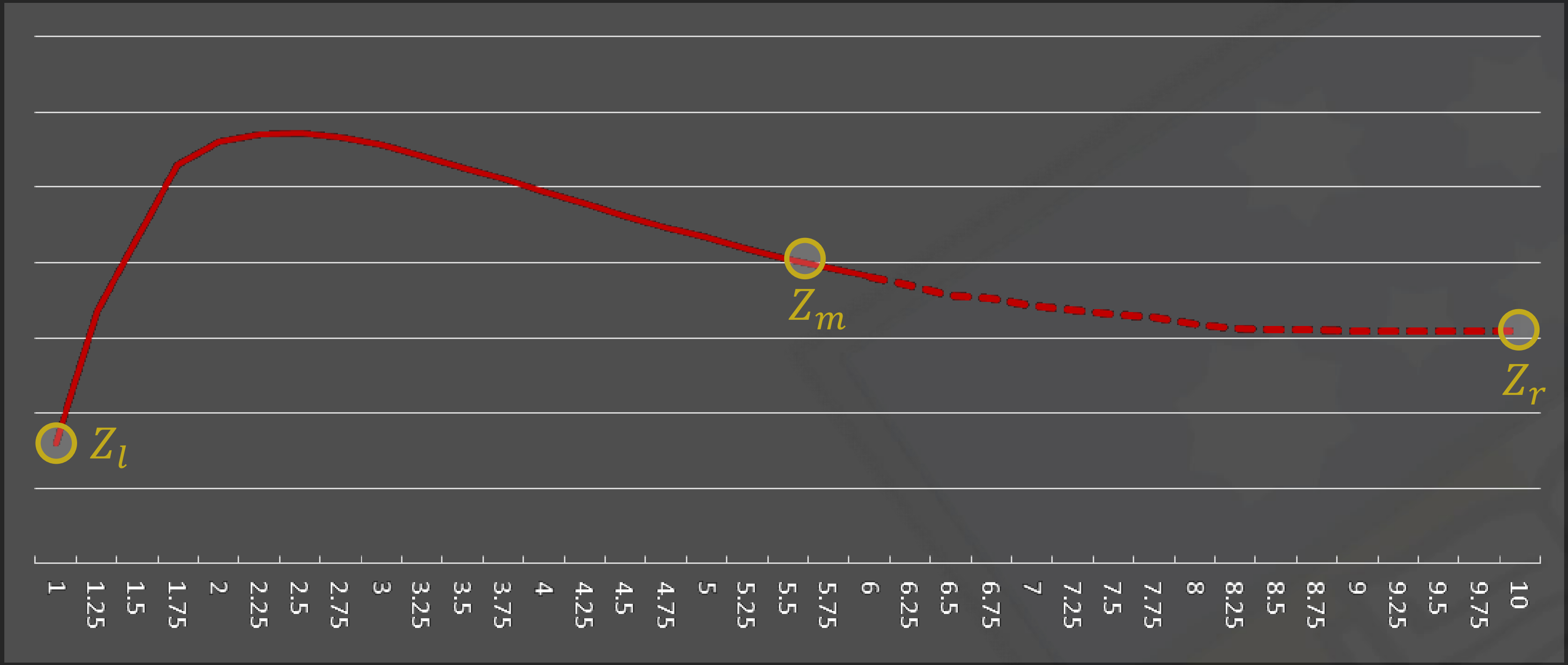


*objective scores averaged over 50 unique tours

THE UNIVERSITY of ADELAIDE

FAST BASIC HEURISTIC

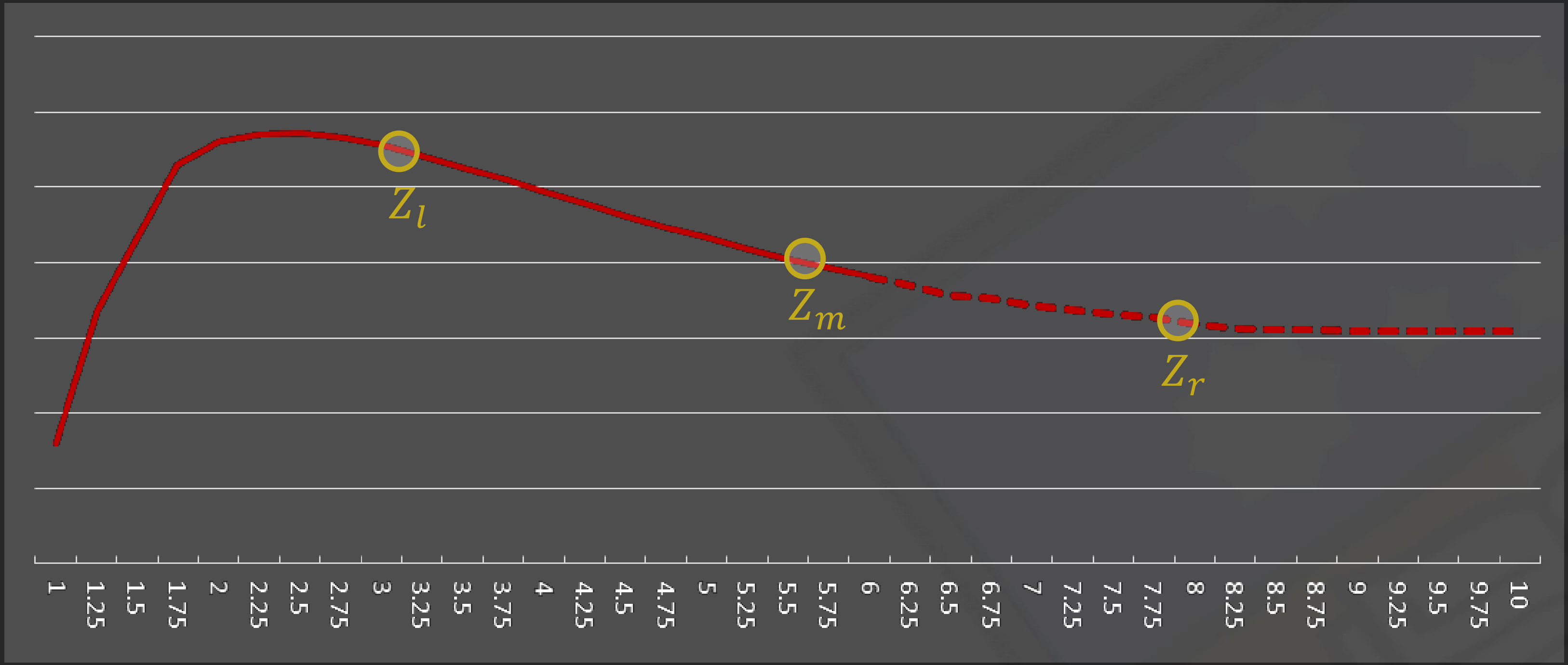
- Sample different α values and compare the objective scores
- Narrow in on the best α value quickly



INSTANCE: rl11849_n118480_uncorr_02

FAST BASIC HEURISTIC

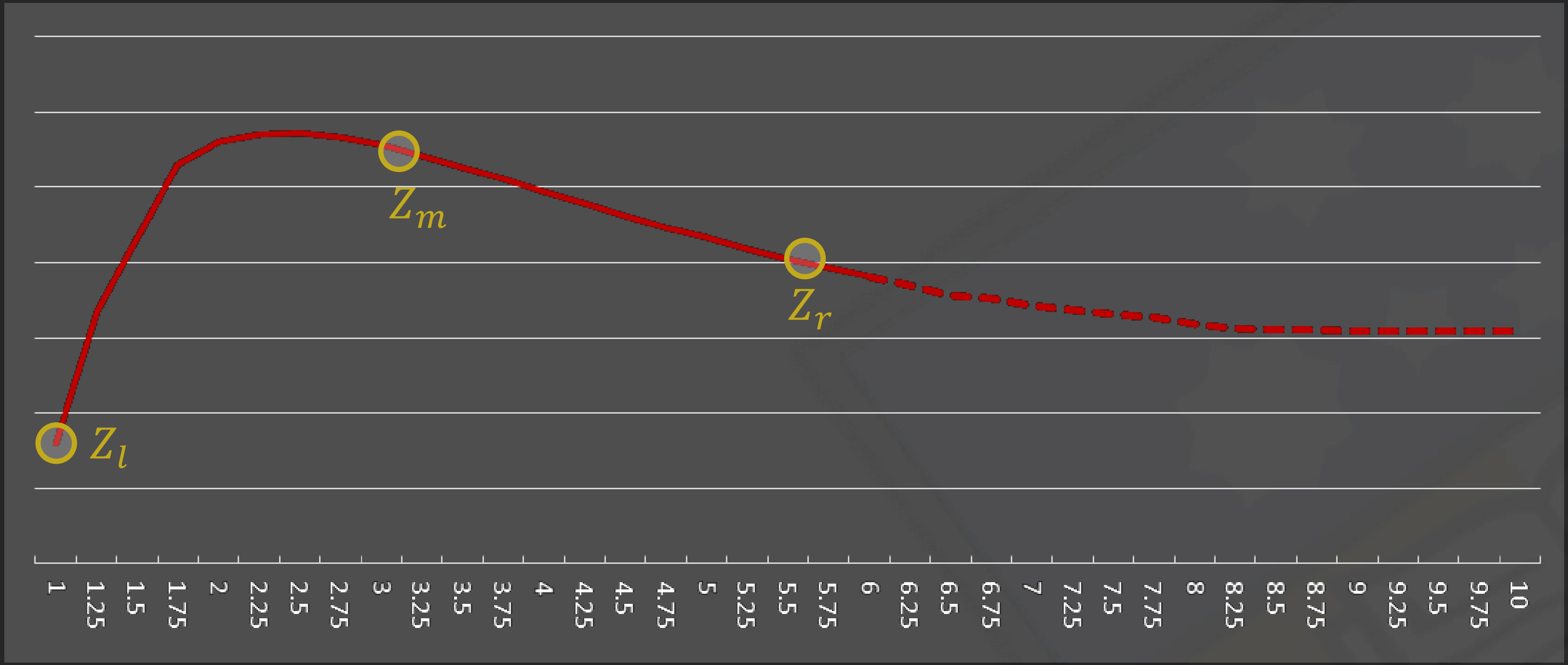
- Sample different α values and compare the objective scores
- Narrow in on the best α value quickly



INSTANCE: rl11849_n118480_uncorr_02

FAST BASIC HEURISTIC

- Sample different α values and compare the objective scores
- Narrow in on the best α value quickly

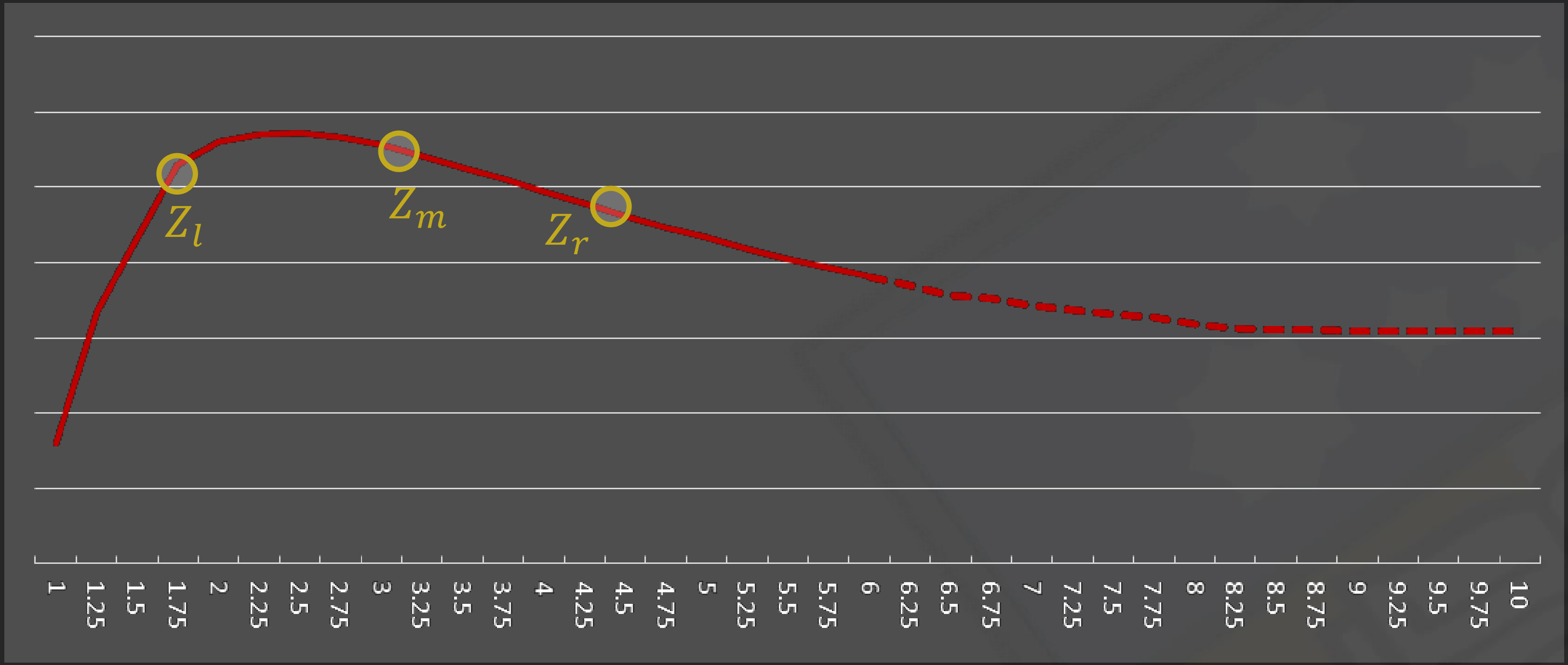


INSTANCE: rl11849_n118480_uncorr_02

THE UNIVERSITY of ADELAIDE

FAST BASIC HEURISTIC

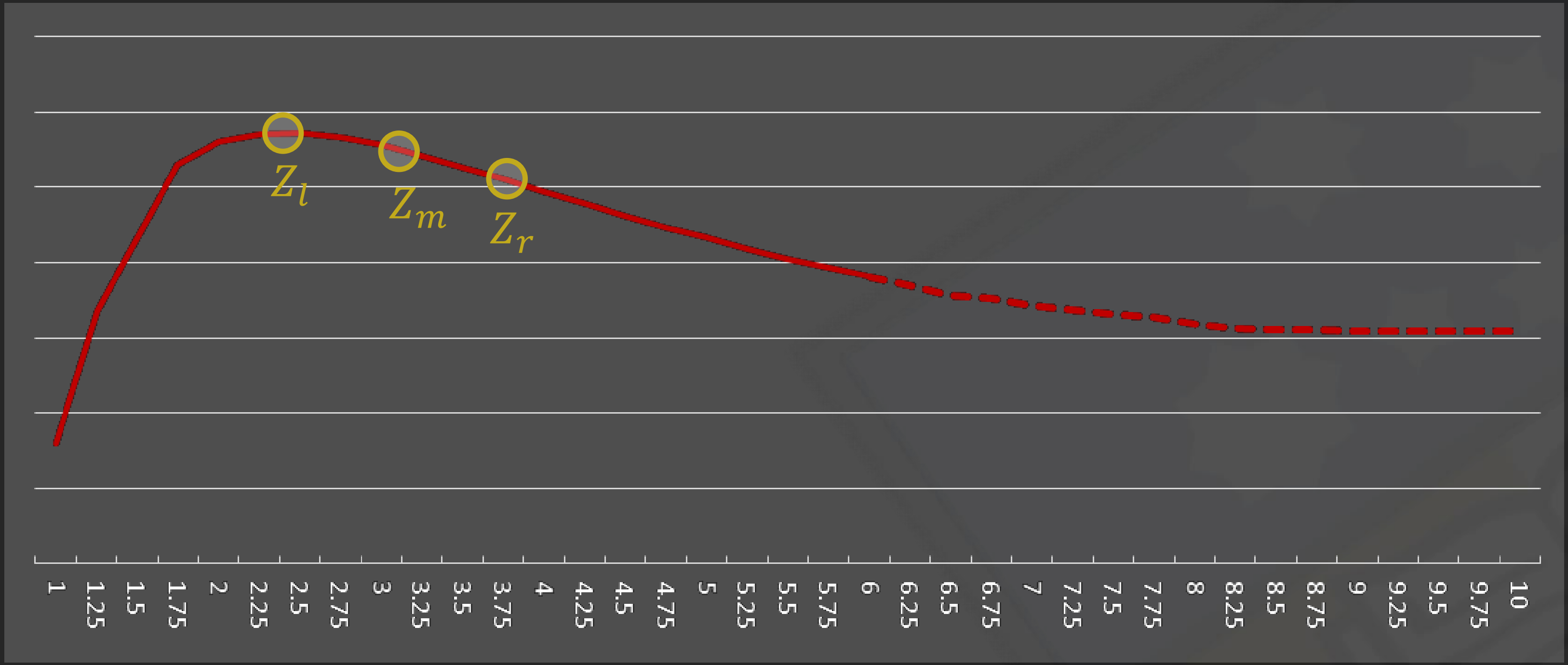
- Sample different α values and compare the objective scores
- Narrow in on the best α value quickly



INSTANCE: rl11849_n118480_uncorr_02

FAST BASIC HEURISTIC

- Sample different α values and compare the objective scores
- Narrow in on the best α value quickly

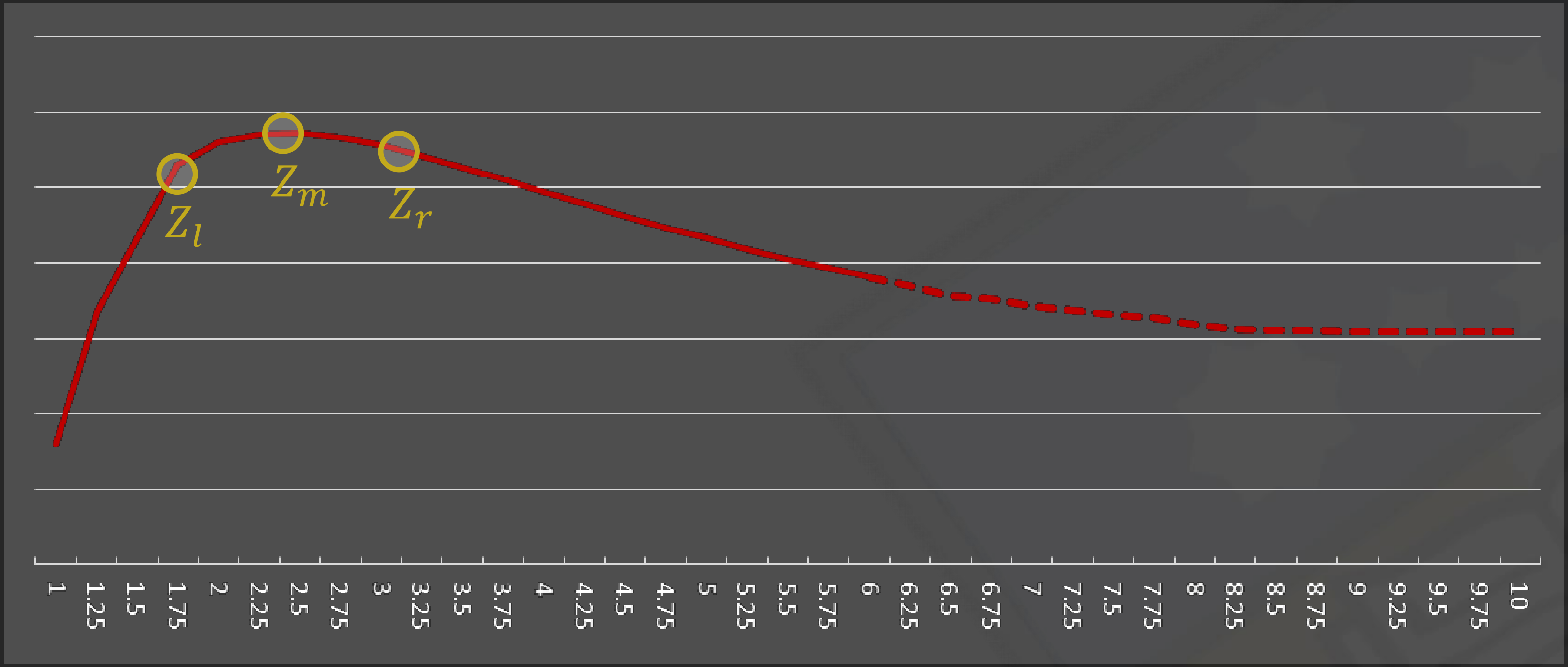


INSTANCE: rl11849_n118480_uncorr_02

THE UNIVERSITY of ADELAIDE

FAST BASIC HEURISTIC

- Sample different α values and compare the objective scores
- Narrow in on the best α value quickly

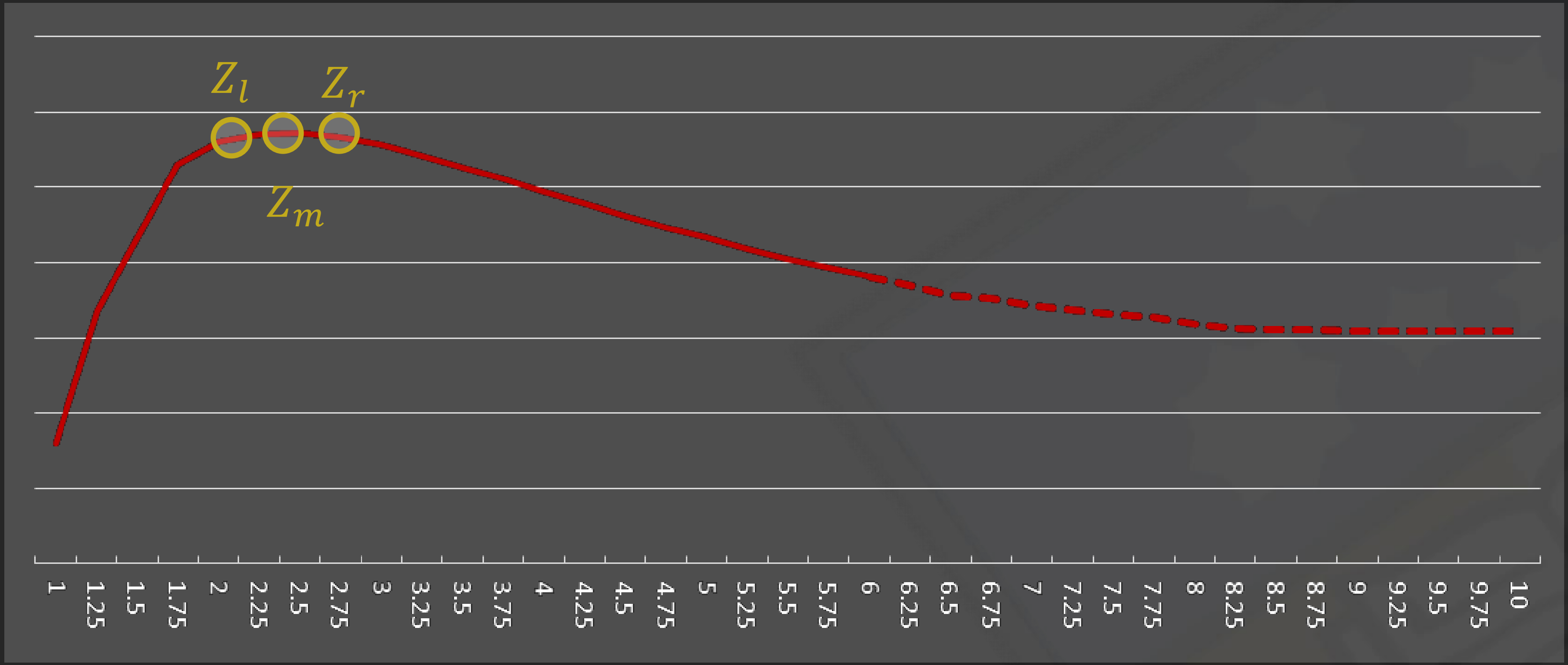


INSTANCE: rl11849_n118480_uncorr_02

THE UNIVERSITY of ADELAIDE

FAST BASIC HEURISTIC

- Sample different α values and compare the objective scores
- Narrow in on the best α value quickly

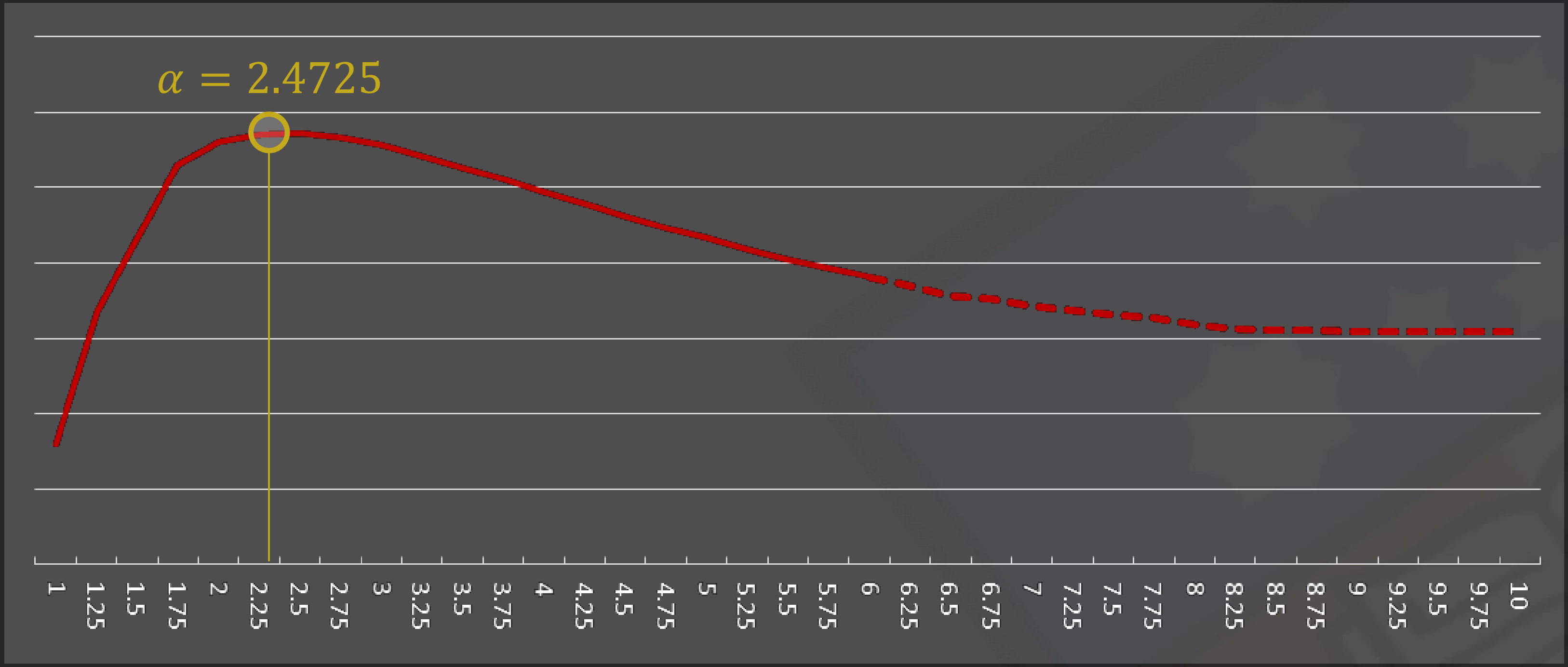


INSTANCE: rl11849_n118480_uncorr_02

THE UNIVERSITY of ADELAIDE

FAST BASIC HEURISTIC

- Sample different α values and compare the objective scores
- Narrow in on the best α value quickly



INSTANCE: rl11849_n118480_uncorr_02

THE UNIVERSITY of ADELAIDE

ADDITIONAL OPERATORS

- The fast basic packing approach is not guaranteed to find globally optimal TTP solution:
  1. Doesn't modify tours based on items
  2. The packing plan it finds may not be optimal for a given tour
- Introduce two local search operations to slightly improve on a given tour and packing plan:
  1. BitFlip – Only modifies packing plan
  2. Insertion – Modifies tour based on provided packing plan

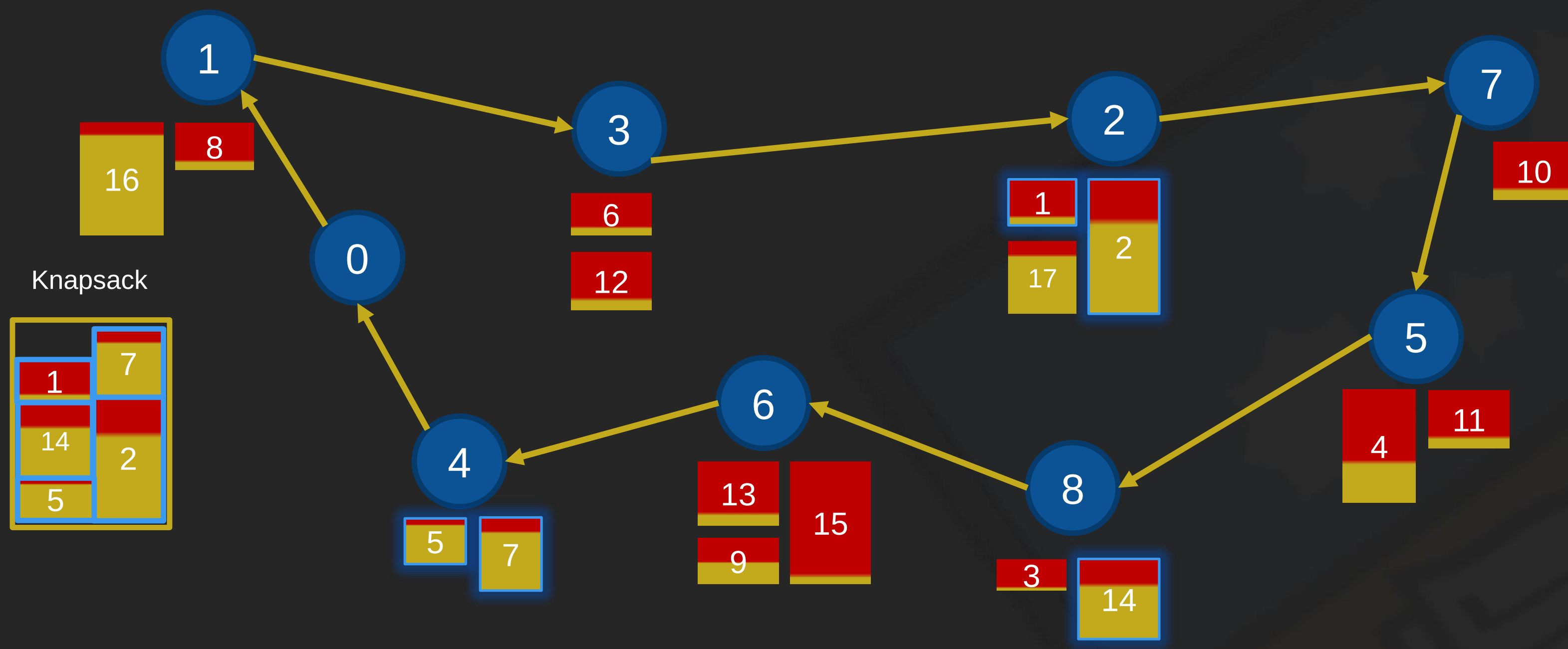
BITFLIP

- Iteratively evaluates the outcome of flipping each bit position corresponding to each item $I_m \in M$ in the packing plan P
- If flipping the bit improves the objective value then the change is kept, otherwise the packing plan is restored
- Can be time consuming on instances with a large number of items as **every** item is checked
- Can be run consecutively a number of times to increase improvements

THE UNIVERSITY of ADELAIDE

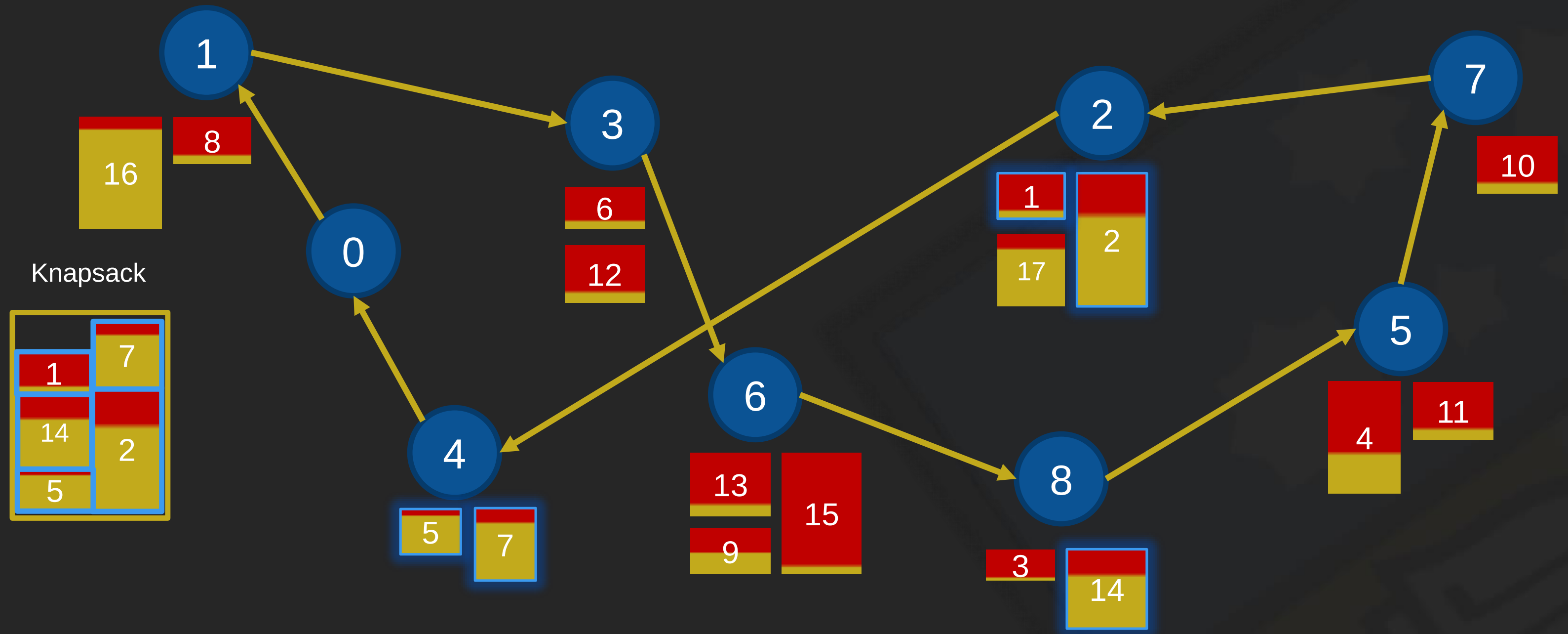
INSERTION

- Takes advantage of situation where valuable item is picked up at a particular city early in the tour and is worth visiting the city later in the tour



INSERTION

- Takes advantage of situation where valuable item is picked up at a particular city early in the tour and is worth visiting the city later in the tour



INSERTION

- Searches over cities in reverse tour order evaluating the effect of inserting each city at all positions before its own in the tour
- If one or more positions are found, the one that achieves the highest objective score is chosen
- Typical good TTP solutions, and solutions constructed by the fast basic heuristic, have many items picked up towards the end of the tours, hence the time consuming **Insertion** operator begins at the end of the tour
- Experiments show **Insertion** makes rare and minor improvements to a TTP solution provided by the fast basic heuristic

ALGORITHM COMBINATIONS

S1: CLK > Fast Packing

S2: CLK > Fast Packing > BitFlip until convergence or time expired

S3: CLK > Fast Packing > (1+1)-EA until convergence or time expired

S4: CLK > Fast Packing > Insertion until convergence or time expired

S5: repeat S1 until time expired

[1] (1+1)-EA is similar to BitFlip however instead of changing every bit which improves the objective score, each bit is changed with a probability $\frac{1}{m}$

CLK: Chained Lin-Kernighan

THE UNIVERSITY of ADELAIDE

ALGORITHM COMBINATIONS

C1: CLK > Fast Packing > repeat one BitFlip then one Insertion until convergence or time expired

C2: CLK > Fast Packing > repeat one BitFlip then one (1+1)-EA then one Insertion until convergence or time expired

C3: Repeat CLK then Fast Packing until 10% of time expired pick best > one BitFlip then one Insertion until time expired

C4: Repeat CLK then Fast Packing until 10% of time expired pick best > one BitFlip then one (1+1)-EA then one Insertion until time expired

C5: repeat C1 until time expired

C6: repeat C2 until time expired

CLK: Chained Lin-Kernighan

THE UNIVERSITY of ADELAIDE

MIP APPROACH

MIP (Mixed Integer Programming) approach of Polyakovskiy, Neumann (2014):

- Given tour able to solve optimal packing plan exactly or approximately
- Very costly in regard to runtimes as it uses a linearization technique to handle non-linear terms in the objective function

EXPERIMENTS

- Compare our algorithm combinations S1-S5 and C1-C6 with the MIP and the MATLS (Memetic Algorithm with the Two-stage Local Search) approach of Mei, Li, Yao (2014)
- Use comprehensive set of benchmark instances from [1,4]:
  - 51 – 85900 cities
  - three types: *uncorrelated*, *uncorrelated with similar weights*, and *bounded strongly correlated*
  - 1,3,5, or 10 items per city for each TSP and KP combination
  - For each TTP configuration there is 10 different instances with varying knapsack capacities

THE UNIVERSITY of ADELAIDE

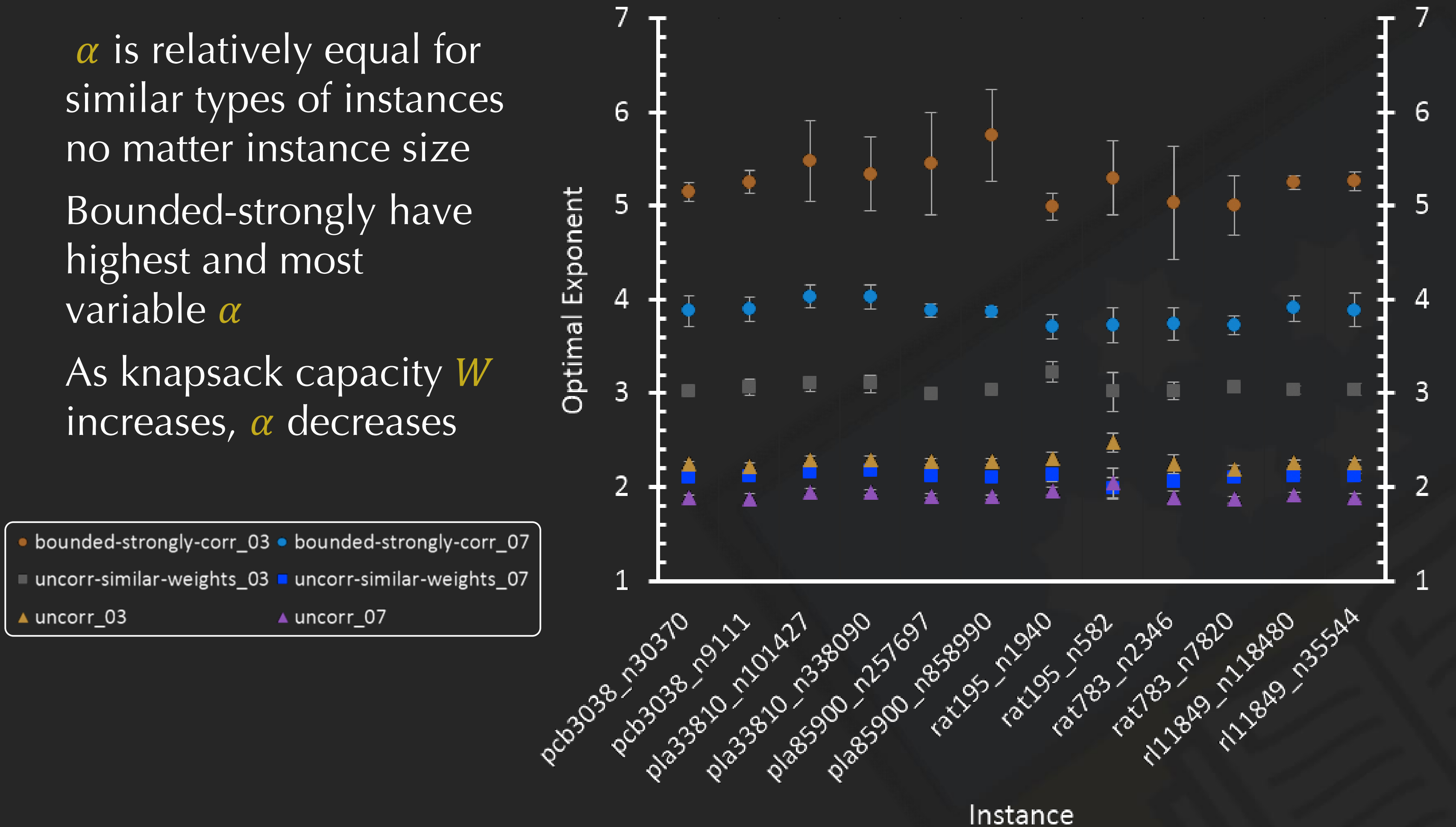
EXPERIMENTS

- From the 9720 benchmark instances 72 representative cases were selected:
  - six different number of cities: *195, 783, 3038, 11849, 33810, 85900*
  - all types: *uncorrelated, uncorrelated with similar weights*, and *bounded strongly correlated*
  - Two different items per city: *3* and *10*
  - Two different knapsack capacities: *3* and *7* times the size of the smallest knapsack
- All algorithms run for 10 minutes per instance, except MIP which ran for 8 hours on instances where $n \in \{33810, 85900\}$
- 30 independent repetitions of algorithms on each instance

THE UNIVERSITY of ADELAIDE

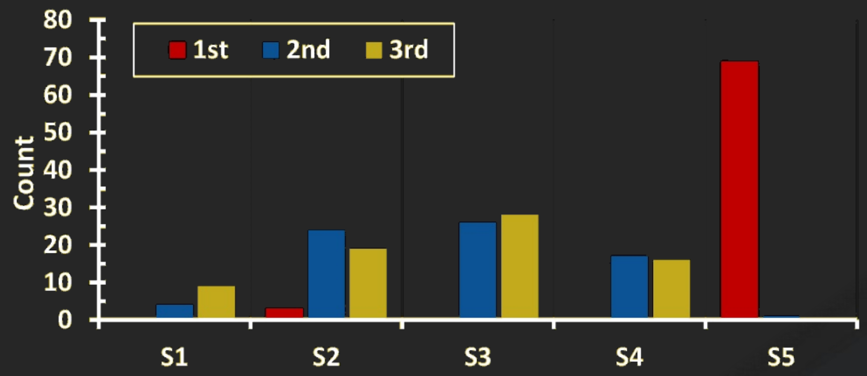
RESULTS

- α is relatively equal for similar types of instances no matter instance size
- Bounded-strongly have highest and most variable α
- As knapsack capacity W increases, α decreases



THE UNIVERSITY of ADELAIDE

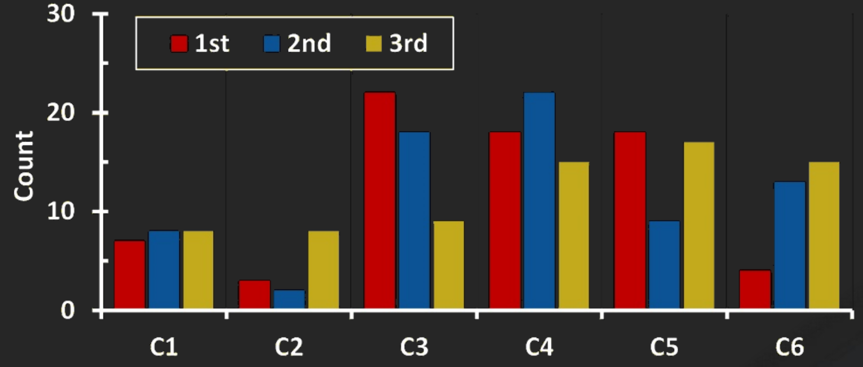
RESULTS



Comparison of the number of first, second, and third placings of S algorithms across the 72 instances

- S5 clearly outperforms the others, showing the importance of a good initial tour
- S2-S4 relatively equal runners up showing they perform on instances where the others do not
- The placings of S4 highlight the necessity to consider modifications to the tour of a TTP solution

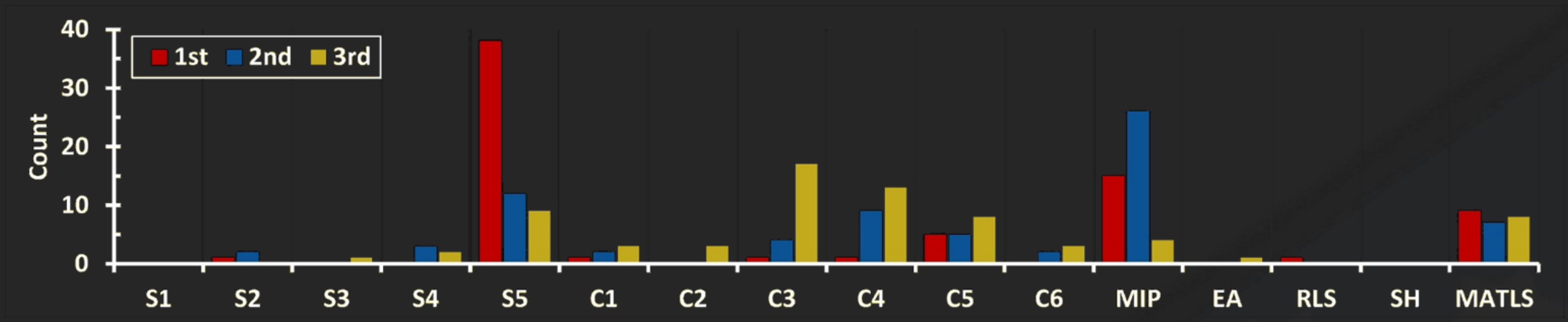
RESULTS



Comparison of the number of first, second, and third placings of C algorithms across the 72 instances

- Recall that C3 and C4 sample several starting tour options, compared to C1 and C2, and that C5 and C6 are the restart variants of C1 and C2
- The dominance C3 and C4 again suggest the importance of finding a good initial TSP tour solution
- C5 and C6 perform better than the single iteration C1 and C2 methods, however they do not perform as well as C3 and C4 which have more time to sample a greater number of initial tours

RESULTS



Comparison of the number of first, second, and third placings of all algorithms across the 72 instances

EA: Evolutionary Algorithm, RLS: Random Local Search, SH: Simple Heuristic. All from [1].

- S5, MIP, and MATLS are the best performing algorithms overall

THE UNIVERSITY of ADELAIDE

RESULTS

n	m	t	F	S1	S5	C3	C4	C5	C6	MIP	MATLS
195	582	bsc	3	88.1	99.6	99.4	99.4	99.6	99.5	100	96.3
			7	88.0	99.2	99.2	99.2	99.2	99.2	99.2	95.6
		unc	3	98.2	99.2	99.3	99.3	99.3	99.3	99.3	99.5
			7	99.2	99.9	100	100	100	100	100	99.8
		usw	3	96.1	98.6	99.0	98.9	99.3	99.0	99.5	98.5
			7	98.2	99.1	99.2	99.2	99.3	99.2	99.3	99.2
	1940	bsc	3	89.6	99.9	99.9	99.9	99.9	99.9	100	98.3
			7	89.2	97.7	97.1	97.5	97.7	97.6	98.1	97.0
		unc	3	96.0	99.1	98.7	98.7	99.3	99.1	99.1	99.0
			7	96.3	98.6	97.7	97.7	98.7	98.5	98.8	99.2
		usw	3	88.4	91.3	91.3	91.6	91.5	91.7	91.4	97.0
			7	92.6	96.3	95.6	95.6	96.4	96.0	96.9	99.4
783	2346	bsc	3	97.8	99.7	99.4	99.4	99.6	99.4	99.7	96.6
			7	95.5	99.3	98.7	98.6	99.1	98.8	99.0	96.6
		unc	3	95.9	98.9	98.5	98.5	98.8	98.5	98.7	98.8
			7	96.3	97.9	97.6	97.7	97.9	97.7	97.8	98.6
		usw	3	95.3	99.1	99.2	99.2	99.5	99.3	99.5	98.7
			7	96.0	99.7	99.6	99.5	99.8	99.5	99.7	98.7
	7820	bsc	3	98.0	99.7	99.6	99.7	99.4	99.5	99.8	94.8
			7	97.0	99.5	99.4	99.4	99.3	99.1	99.5	94.5
		unc	3	96.8	99.3	99.2	99.1	99.2	98.7	99.3	98.5
			7	97.9	99.5	99.4	99.3	99.3	99.1	99.4	99.0
		usw	3	96.0	99.3	99.6	99.6	99.4	99.0	99.6	97.7
			7	95.9	99.3	99.2	99.2	99.2	98.4	99.4	98.1
3038	9111	bsc	3	97.9	99.5	99.1	99.1	98.1	98.0	99.1	93.9
			7	97.6	99.4	99.0	99.0	97.7	97.4	98.8	94.4
		unc	3	98.0	99.7	99.4	99.4	98.5	98.0	99.5	99.0
			7	98.3	99.7	99.5	99.4	98.9	98.5	99.6	99.4
		usw	3	97.0	99.1	99.2	99.1	97.2	97.2	99.2	98.2
			7	97.6	99.6	99.3	99.2	98.4	97.7	99.3	99.0
	30370	bsc	3	98.1	99.6	99.3	99.3	99.0	99.0	99.3	96.7
			7	97.0	99.2	98.7	98.8	98.6	98.2	98.9	95.8
		unc	3	97.1	99.6	99.2	99.1	98.9	98.8	99.3	99.0
			7	97.8	99.5	99.3	99.3	99.2	98.9	99.3	99.2
		usw	3	94.8	98.9	98.2	98.3	97.6	97.6	98.6	98.3
			7	96.2	99.1	98.6	98.4	98.5	97.9	98.6	98.6

n	m	t	F	S1	S5	C3	C4	C5	C6	MIP	MATLS
11849	35544	bsc	3	97.1	99.2	98.4	98.6	97.3	97.4	97.5	93.5
			7	96.7	98.9	97.9	98.1	96.6	96.7	96.7	93.9
		unc	3	97.4	99.0	98.4	98.5	97.6	97.8	98.0	98.4
			7	97.9	99.5	98.9	99.0	98.0	98.3	98.4	99.3
		usw	3	96.3	98.6	97.8	98.0	96.2	96.5	97.2	97.6
			7	97.1	99.0	98.5	98.5	97.0	97.3	97.3	98.7
	35544	bsc	3	96.7	99.0	98.4	98.3	96.9	97.0	97.9	93.6
			7	96.2	99.2	98.1	98.3	96.2	96.4	97.4	94.0
		unc	3	97.2	99.2	98.6	98.7	97.4	97.6	98.3	98.4
			7	97.4	99.2	98.6	98.4	97.9	97.8	98.5	98.8
		usw	3	95.3	98.3	97.8	97.8	95.6	95.9	97.1	97.6
			7	96.2	98.9	98.1	98.0	96.7	96.7	97.9	98.5
33810	101427	bsc	3	91.3	97.9	95.5	94.0	93.9	92.0	98.3	94.4
			7	91.0	97.9	94.2	95.3	93.8	91.7	96.5	94.1
		unc	3	70.6	73.5	71.4	71.5	71.2	70.9	73.3	75.8
			7	95.1	98.2	96.4	96.0	95.3	95.5	99.9	98.4
		usw	3	90.4	97.5	93.7	93.3	92.5	91.8	96.2	95.9
			7	92.2	98.0	94.7	93.9	93.9	93.5	98.1	97.4
	338090	bsc	3	92.2	97.3	93.8	93.3	92.5	92.4	99.1	93.9
			7	92.6	97.1	94.9	94.3	92.6	93.3	96.9	94.6
		unc	3	94.7	98.3	95.3	95.5	95.8	95.0	97.8	98.0
			7	95.0	98.4	96.0	96.1	96.2	95.7	98.5	98.7
		usw	3	91.3	97.7	93.5	92.8	92.1	92.3	98.3	96.4
			7	93.9	98.3	94.4	95.1	94.1	94.6	99.5	98.3
85900	338090	bsc	3	95.8	98.3	96.3	95.8	95.9	96.4	97.6	-
			7	96.1	97.8	96.8	95.9	96.3	97.1	98.4	-
		unc	3	97.4	98.9	94.2	94.1	97.8	97.7	-	-
			7	97.6	98.6	95.6	95.4	98.0	98.2	98.1	-
		usw	3	95.1	97.6	92.6	92.3	96.1	95.2	-	-
			7	95.8	97.4	93.4	93.3	96.4	96.0	97.9	-
	858990	bsc	3	95.9	96.8	96.2	97.1	95.6	96.3	-	-
			7	96.6	97.2	97.0	97.6	96.6	96.3	-	94.1
		unc	3	97.6	97.5	92.1	92.0	97.6	97.3	-	-
			7	97.9	97.9	94.6	94.5	97.8	97.7	-	98.5
		usw	3	95.7	97.5	91.8	92.7	96.3	95.3	-	-
			7	96.6	96.9	93.2	93.2	96.5	96.3	-	97.9
avg				95.2	98.2	97.0	96.9	96.9	96.8	87.2	84.9
avg^{-85900}				95.0	98.3	97.5	97.4	97.0	96.8	98.1	97.0

THE UNIVERSITY of ADELAIDE

Objective value ratios compared to maximum found across all algorithms and iterations

CONCLUSIONS

- The strength of our fast basic packing heuristic is its speed, allowing more initial Lin-Kernighan tours to be sampled *(15-60 milliseconds for 195 cities, and 18-110 seconds for 85,900 cities)*
- Local search operators such as BitFlip and Insertion have positive yet limited effect due to their computational complexity
- Even MIP approach only just achieves comparable performance with the limited time availability

THE UNIVERSITY *of* ADELAIDE

REFERENCES

1. S. Polyakovskiy, M. R. Bonyadi, M. Wagner, Z. Michalewicz, and F. Neumann. A comprehensive benchmark set and heuristics for the traveling thief problem. In Genetic and Evolutionary Computation Conference (GECCO), pp. 477{484. ACM, 2014
2. S. Polyakovskiy and F. Neumann. Packing while traveling: Mixed integer programming for a class of nonlinear knapsack problems. CoRR, abs/1411.5768, 2014
3. Y. Mei, X. Li, and X. Yao. Improving efficiency of heuristics for the large scale traveling thief problem. In Simulated Evolution and Learning (SEAL), Vol. 8886 of LNCS, pp. 631{643. Springer, 2014
4. TTP Test Data. See http://cs.adelaide.edu.au/~optlog/research/ttp.php