Supplementary Material: Qubit Routing using Graph Neural Network aided Monte Carlo Tree Search

Author Placeholder

Affiliation Placeholder

A MCTS Algorithm

The following is the detailed implementation of our MCTS procedure. In each iteration of the Monte Carlo Tree Search, we start at the root, keep selecting nodes unless we find that we have selected a node never seen before, expand it, and propagate the resultant rewards up the tree to its ancestors.

In our implementation, the value of the decay factor γ is different for the COMMIT actions from those of SWAP actions. We use a decay of 1.0 (i.e. no decay) for the non-commit actions and 0.95 for commit actions. So we only have decay in reward propagation across 2 different states, not within the construction of a single action. The function step(s, a) is a call to the environment to schedule the gates as described by the action a and evolve the state $s_t \xrightarrow{a} s_{t+1}$.

B Results on Google Sycamore

Following is the plot of the average Circuit Depth ratio produced by our method on the Google Sycamore processor. Sycamore has a much larger size of 53 qubits. Our method manages to give an average depth ratio of 1.64 here, and is the best of all competing routing methods.

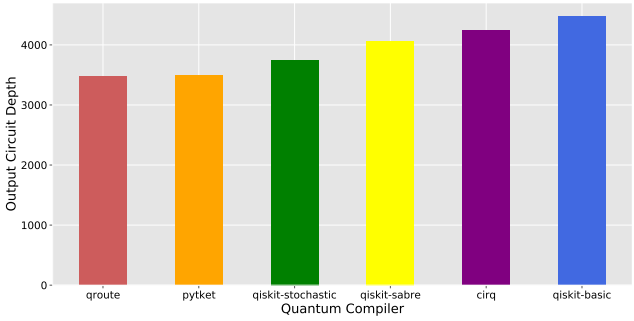


Figure 1: A comparative of the performance of the different routing methods on the small circuit dataset when routing on Google Sycamore device.

Algorithm 1: Monte Carlo tree search

Data: state s_t
Initialize: root ← (s_t, empty action set)
loop *n_mcts times*
  (s, a) ← root
  repeat
    Compute **UCT** values using prior + noise
    Select *move* that maximizes UCT value
    if *(s, a).child[move]* ≠ *null* **then**
      (s, a) ← (s, a).child[move]
    else
      if *move* = *COMMIT* **then**
        s' ← **step** (s, a)
        a' ← empty set
      else
        s' ← s
        a' ← **insert** into a the qubit pair corresponding to the move
      end
      state.child[move] ← (s', a')
      store reward[(s, a), move] ← $\mathcal{R}(s', a')$ - $\mathcal{R}(s, a)$
    end
  until *last taken move was* ***expand***
  reward ← **evaluation** from model of (s, a)
  while *(s, a)* ≠ *root* **do**
    p-move ← move from parent of (s, a) to (s, a)
    (s, a) ← parent in tree of (s, a)
    reward ← reward[(s, a), p-move] + γ· reward
    update (s, a).Q-value[move] with reward
    increment (s, a).N-value[move] by 1
  end
end
memorize the Q-values and N-values at the root for training the model later
(s, a) ← root
repeat
  (s, a) ← child of (s, a) with maximum Q-value
until *move* ≠ *COMMIT*
return a

C Example of Routing Process

In this section, we show an example run of our algorithm on a 3×3 device with a normal grid topology, i.e. only qubits adjacent to each other are connected. In the images that follows, we have shown the evolution of the state, the value of the state at each timestep and the action, i.e. the set of gates which are being scheduled. Our MCTS is also responsible for constructing each action by putting together several moves (which are either adding individual gates to the action, or a committing the action for this timestep), that process is not demonstrated in the images. A point to note is that at the start of each timestep, the locks on all qubits may not necessarily be open, because there can be operations which were scheduled in a previous timestep and span over several timesteps. However, this is not the case in our example here where all the gates are assumed to take the same amount of time. We have provided a video simulation of this evolution as a supplementary, as well as the code to visualize this for other circuits.

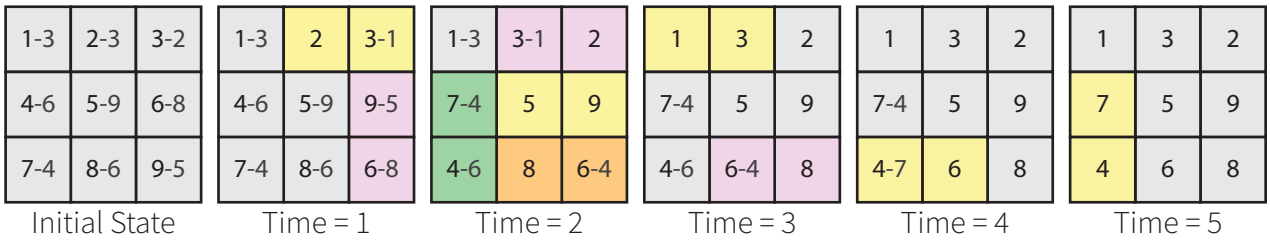
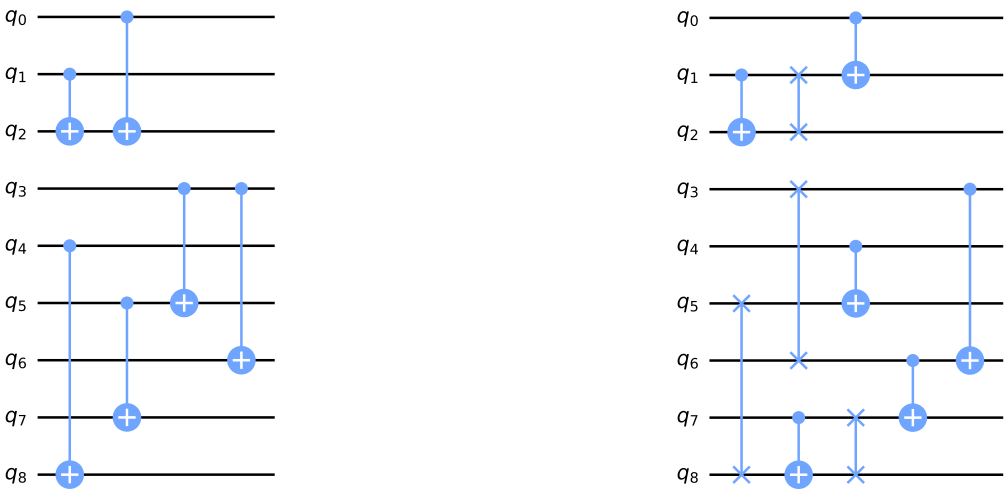


Figure 2: The step by step evolution of the state as the circuit is getting routed. The state is shown on a 3×3 grid, where in each cell we have the node ID and the next node that it need to participate in a 2-qubit operation with. The yellow and orange colors represent that those 2 qubits have participated in a 2 qubit operation like CNOT, which was scheduled in the previous timestep. The green and purple colors represent that they have just participated in a SWAP operation. Any qubit which is colored was locked in the previous timestep when the action that scheduled it was getting constructed. At time=5, the circuit has been scheduled and none of the qubits have any targets left.



(a) Quantum circuit

(b) Decomposed circuit

Figure 3: This figure shows the input and output of the routing process shown above in Figure 2. The input circuit was used to decide the targets of the qubits. Gates are added to the output circuit whenever a 2-qubit operation, whether CNOT or SWAP is applied by the router. We can check that both these circuits are equivalent

.

D Tabulated Results

D.1 Random Test Circuits

Table 1: **Comparative results for a set of randomly generated test circuits**

Input Circuit		Output Circuit Depth					
Number of Gates	Number of Layers	Qroute	Cirq	Qiskit (basic)	Qiskit (stochastic)	Qiskit (sabre)	t|ket⟩
30	11	20	29	31	21	24	19
30	11	19	36	39	23	27	28
30	10	22	32	28	23	23	23
30	8	18	24	32	20	33	26
30	7	17	19	35	17	23	30
30	11	18	39	34	22	31	26
30	10	19	22	34	21	20	26
30	9	17	24	31	21	31	33
30	10	17	24	36	22	31	21
30	10	21	23	39	22	27	23
50	18	31	57	66	41	50	46
50	17	29	54	63	37	48	46
50	12	34	47	62	31	50	53
50	17	28	62	67	38	38	58
50	17	33	42	61	39	50	51
50	15	40	49	61	38	48	41
50	18	35	60	66	38	52	50
50	18	33	54	53	35	42	52
50	13	32	58	63	31	39	35
50	16	30	52	59	37	44	42
70	19	39	85	93	45	59	76
70	21	47	96	71	41	56	60
70	18	46	64	81	43	57	67
70	21	59	83	84	53	58	79
70	18	47	67	60	44	55	80
70	21	45	77	83	46	69	59
70	19	41	63	76	44	52	74
70	17	40	68	67	42	62	63
70	23	37	70	84	52	63	72
70	23	40	64	91	49	73	60
90	29	53	106	93	64	80	114
90	26	64	103	117	64	73	77
90	28	56	93	111	64	85	89
90	22	57	94	114	54	75	87
90	32	58	99	108	66	87	84
90	23	54	104	127	60	97	90
90	28	52	96	103	60	80	92
90	25	50	97	113	60	76	75
90	23	51	103	107	54	74	82
90	25	56	96	111	61	79	91
110	34	63	133	113	72	97	137
110	27	65	128	143	70	95	118
110	31	64	117	128	69	95	126
110	30	73	116	139	66	104	106
110	32	62	108	129	75	100	124
110	36	68	112	135	78	93	107
110	33	94	135	158	74	99	103
110	33	67	124	110	75	96	117
110	31	64	114	129	71	101	113
110	30	65	116	145	69	97	115

Table 1 continued from previous page

Input Circuit		Output Circuit Depth					
Number of Gates	Number of Layers	Qroute	Cirq	Qiskit (basic)	Qiskit (stochastic)	Qiskit (sabre)	t|ket⟩
130	33	74	151	149	74	113	154
130	33	91	135	166	79	122	126
130	38	77	130	162	91	123	133
130	32	77	112	153	75	116	139
130	38	71	145	151	94	113	137
130	34	66	127	153	79	98	122
130	35	75	131	151	89	101	144
130	31	70	114	157	74	107	135
130	33	76	130	141	79	102	128
130	41	95	148	161	91	102	114
150	35	87	175	151	86	109	142
150	44	92	194	195	104	154	158
150	38	84	162	177	93	136	149
150	35	79	128	178	84	123	149
150	48	96	177	195	101	138	158
150	43	92	179	167	97	126	142
150	41	90	171	185	98	120	165
150	39	85	155	158	91	125	158
150	39	88	148	182	94	135	158
150	38	89	178	162	96	123	159

D.2 Small Realistic Circuits

Table 2: **Comparative results for low-depth realistic test circuits**

Input Circuit		Output Circuit Depth						
Circuit Name	Layers	DQN (Estimate)	Qroute	Cirq	Qiskit (basic)	Qiskit (stochastic)	Qiskit (sabre)	t|ket⟩
4gt11_83	14	17	16	22	18	19	18	15
decod24-v0_38	23	28	30	43	23	31	32	24
alu-v3_34	23	28	27	39	28	28	25	28
decod24-v3_45	57	68	72	79	74	84	81	77
4gt4-v0_80	71	85	89	108	111	91	109	128
alu-v0_27	15	18	16	19	21	19	17	17
miller_11	23	28	25	23	36	36	34	35
4gt11_82	18	22	19	28	22	24	23	24
mod10_176	70	84	83	113	87	94	87	96
ex1_226	5	6	7	8	10	7	8	6
4gt5_75	33	40	38	54	40	41	46	43
ising_model_10	20	24	23	40	20	20	20	5
4gt11_84	8	10	8	13	11	11	12	8
4mod5-v0_18	31	37	34	53	39	40	40	33
alu-v4_37	16	20	20	16	23	22	25	24
qft_10	34	41	53	81	113	50	75	49
4mod5-v0_19	15	18	17	25	20	19	24	26
alu-v0_27_example	15	18	16	18	21	20	17	17
ex-1_166	9	11	13	12	14	12	11	13
4mod7-v1_96	65	78	75	91	83	97	85	88
4mod5-v1_22	10	12	12	17	13	12	12	14
4gt12-v1_89	88	105	109	163	126	135	134	125
alu-v1_29	15	18	16	21	19	20	21	19
mod5d2_64	25	30	30	45	30	38	34	31
4mod7-v0_94	66	79	77	131	83	92	92	82
4gt13_91	46	55	53	64	52	68	60	52

Table 2 continued from previous page

Input Circuit		Output Circuit Depth						
Circuit Name	Layers	DQN (Estimate)	Qroute	Cirq	Qiskit (basic)	Qiskit (stochastic)	Qiskit (sabre)	t|ket⟩
4mod5-v0_20	9	11	11	16	17	10	10	10
alu-v2_33	15	18	19	26	17	19	20	17
4_49_16	91	109	99	138	107	129	104	128
decod24-v2_43	22	27	25	38	26	28	33	25
4gt10-v1_81	60	72	71	113	82	80	81	75
alu-bdd_288	35	42	47	60	55	52	54	44
4mod5-v1_23	30	36	33	46	45	45	48	40
one-two-three-v2_100	29	35	35	40	41	41	40	39
rd53_138	42	50	54	70	67	69	78	59
alu-v2_32	64	77	72	96	88	87	98	96
rd32_270	35	42	38	53	48	53	49	40
aj-e11_165	63	75	73	103	82	90	81	82
4gt12-v0_88	77	92	90	128	116	111	107	140
decod24-v1_41	35	42	38	55	42	50	47	43
3_17_13	17	21	24	17	26	18	24	22
4mod5-v0_19	16	20	17	27	16	21	22	13
mini_alu_305	53	64	63	113	86	81	79	81
one-two-three-v0_98	59	71	71	82	69	81	77	87
4gt13_90	50	60	54	72	56	63	56	80
4mod5-bdd_287	31	37	41	48	35	48	50	35
ham3_102	11	14	14	16	15	16	15	9
alu-v1_28	16	20	16	21	20	18	22	18
rd32-v0_66	16	20	19	20	20	20	20	14
cnt3-5_179	43	52	64	91	72	61	65	83
4gt13_92	26	31	30	41	33	38	36	29
alu-v4_36	47	56	55	59	65	68	61	60
rd32-v1_68	16	20	17	21	20	20	20	14
4gt13-v1_93	27	33	29	34	35	36	36	31
4gt5_76	42	50	47	69	53	52	53	53
mod5d1_63	11	14	14	17	12	12	14	14
graycode6_47	5	6	5	9	5	5	5	5
xor5_254	5	6	5	8	10	8	8	6
decod24-bdd_294	31	37	34	50	40	40	46	37
alu-v0_26	35	42	41	62	47	48	45	54
mod5mils_65	16	20	19	21	17	21	25	18
alu-v3_35	16	20	20	25	23	21	21	24
one-two-three-v1_99	56	67	60	87	70	84	70	95
one-two-three-v3_101	29	35	34	43	36	37	38	46
4gt5_77	51	61	61	78	63	66	70	66

D.3 Large Realistic Circuits

Table 3: **Comparative results for long-depth realistic test circuits**

Input Circuit		Output Circuit Depth				
Circuit Name	Number of Gates	Qroute	t|ket⟩	Qiskit (basic)	Qiskit (stochastic)	Qiskit (sabre)
rd84_142	154	120	154	142	138	133
adr4_197	1498	1580	1770	1840	1968	1988
radd_250	1405	1504	1799	1812	1815	1888
z4_268	1343	1400	1670	1623	1718	1914
sym6_145	1701	1806	2167	2168	2261	2299
misex1_241	2100	2231	2580	2770	2681	2944
rd73_252	2319	2468	2793	2943	3071	3132

Table 3 continued from previous page

Input Circuit		Output Circuit Depth				
cycle10_2_110	2648	2941	3380	3418	3485	3705
square_root_7	3089	3327	4560	3759	3822	3695
sqn_258	4459	4779	5535	5526	5696	6252
rd84_253	5960	6264	7507	7411	7537	8843