SOLO: Search Online, Learn Offline for Combinatorial Optimization Problems

Joel Oren[1], Chana Ross[1], Maksym Lefarov[1], Felix Richter[1], Ayal Taitler[2], Zohar Feldman[1], Dotan Di Castro[1], Christian Daniel[1]

[1] Bosch Center for Artificial Intelligence

[2] Technion – Israel Institute of Technology

TECHNION Israel Institute of Technology

BCAI Bosch Center For AI

August 2021

INTRODUCTION

A combinatorial optimization (CO) problem is given by $\langle \mathcal{I}, \mathcal{S}, f \rangle$ where:

- $\mathcal{I}$ – is the set of problem instances
- $\mathcal{S}$ – maps an instance $I \in \mathcal{I}$ to its set of feasible solutions
- f – objective function mapping solutions in $S(I)$ to real values

Parallel Machine Scheduling Problem (PMSP)

Specifically the unrelated machines scheduling with setup and processing time.

- m – number of machines
- n – number of jobs
- $p_{i,j}$ – processing time of job i on machine j
- $s_{i,l}$ – setup time to pass if job of class i is to be processed after job of class l
- w_i – weight of job i

Objective: minimize sum of weighted completion times

Capacitated Vehicle Routing problem (CVRP)

Specifically the single vehicle, single commodity routing problem.

- N – number of customers
- C^* – vehicle capacity
- d_i – demand of customer i, $d_i \le C^*$
- o – commodity location
- p_i – customers locations

Objective: total distance traveled by the vehicle

Offline variant: everything is known a-priori, e.g., all jobs are in the system (PMPS)

Online variant: dynamic arrivals of assigned variables, e.g., jobs (PMPS)

MODELING

Setting

A CO problem is modeled by a sequential decision process, specifically finite horizon Markov Decision Process (MDP) $\langle S, A, T, R \rangle$.

Event-based process

Decision Points: event, a change in the system, i.e., job arrival\machine is free (PMPS), vehicle reaches a destination\customer arrival (CVRP).

- S – is the set of states, i.e., all the entities (jobs, machines, customers) and their properties, (size may change!)
- A – partial variables assignment, e.g., assign job j to machine i.
- T – dynamics of the process correlated to the passed time.
- R – reward, i.e., minus the cost of the time passed between last two events, incurred by taking action $a_t \in A$ at decision point t.

Graph Encoding

Mapping from state space to graph space representation, each state is a graph!

$$s_t \in S \quad \rightarrow \quad \zeta(s_t) = G = \langle V, E, f^v, f^e, f^g \rangle$$

- V – is the set of vertices, the entities in the problems, e.g., machines, jobs, customers, etc.
- E – the set of edges connecting between the vertices, represents relation and information flow.
- f^v, f^e, f^g – features of the vertices, edges and graph respectively.

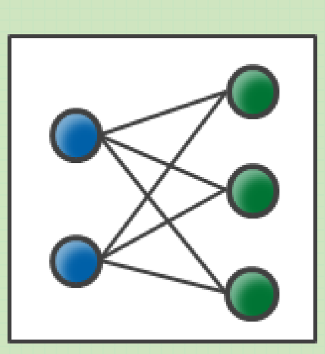

Figure 1: The GNN representation of PMPS. Bi-partite graph. Edges represents possibility of scheduling a job on a machine.
● vehicle ● customer

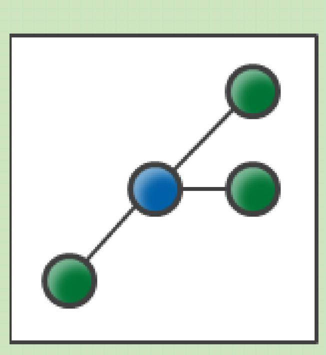

Figure 2: The graph representation of CRVP. Star-graph. Edges represents possible route of the vehicle to a customer.
● vehicle ● customer

Actions Corresponds directly to the graph edges

job features					machine features	
p_j	w_j	a_j	$h^c_{\mathcal{K}_{u_j}}$	$\mathbb{I}^{node}_{u_j}$	0	0
0	0	0	0	$\mathbb{I}^{node}_{v_i}$	r_i	$h^{c+1}_{\mathcal{K}_{v_i}}$

Figure 3: node feature vector of PMPS. Unified representation for all node types.

METHOD

Learn Offline

Deep Reinforcement Learning (DRL) using Deep Q-networks (DQN).

- Simulate problems of *different* sizes (enabled by the graph representation)
- Learn size agnostic scheduling policy.
- Generalize to problems larger than simulated.

Search Online

Monte Carlo Tree Search (MCTS) to reduce erroneous assignment (small perturbations have great effect on the objective in CO).

- Use learned $\tilde{Q}$-Net from offline stage as heuristic.
- Action pruning, choose only between actions with the top k $\tilde{Q}$-Net values.
- Suppress future arrivals after some ΔT

Theoretical optimality is compromised for better empirical results

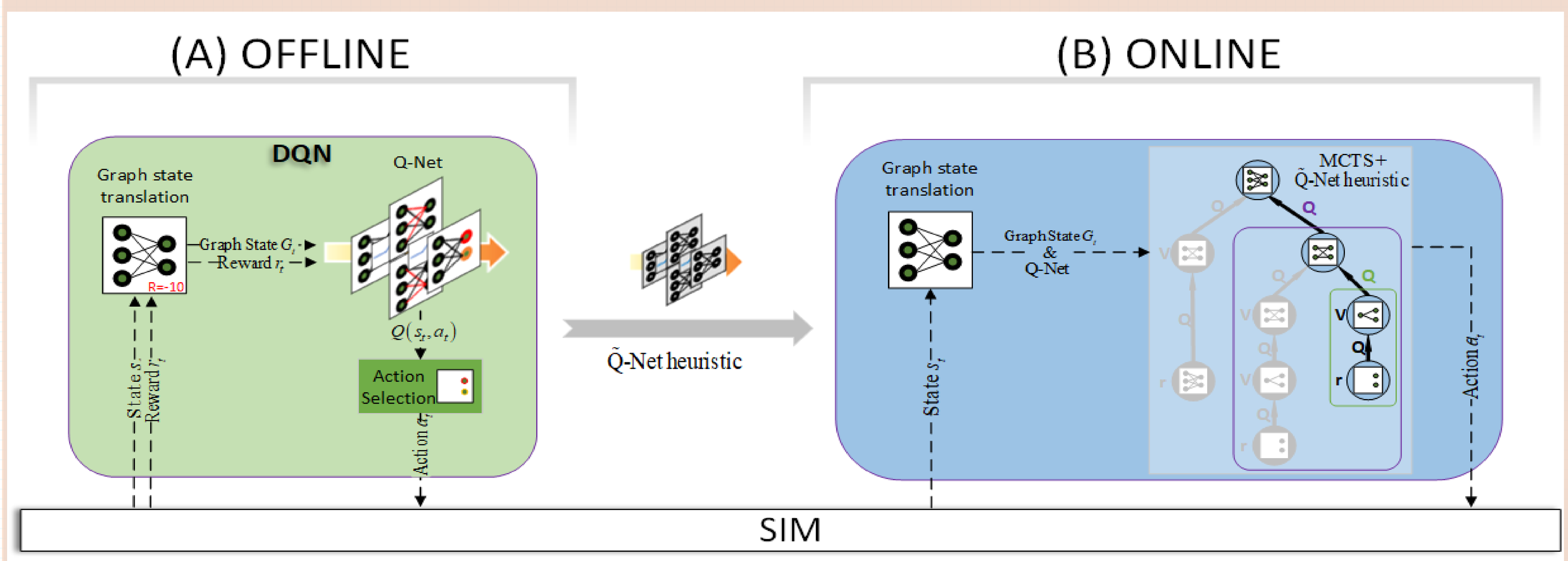

Figure 4: A schematic overview of SOLO. On the left, a depiction of our DQN training process, which produces the $\tilde{Q}$-Net heuristic. On the right is our planning procedure that, for each step, runs our modified MCTS with $\tilde{Q}$Net as a heuristic

EMPIRICAL EVALUATION

Offline PMSP	liao 20	liao 80
WSPT	-16570.16 (5.82%)	-182357.02 (4.15%)
CPLEX	-15658.46 (0%)	-175084.88 (0%)
NeuralRewriter	-16540.28 (5.63%)	-182450.02 (4.21%)
Q-net	-15906.32 (1.58%)	-178444.74 (1.92%)
MCTS+WSPT	-15876.88 (1.39%)	-176439.74 (0.77%)
SOLO	-15695.94 (0.24%)	-175524.34 (0.25%)
SOLO+Prune	**-15683.46 (0.16%)**	**-175164.58 (0.05%)**
optimal	-15628.68 (-0.19%)	

Online PMSP	3 machines	10 machines
WSPT	-40601.34 (15.04%)	-29102.5 (18.87%)
CPLEX	-35294.38 (0%)	-24481.9 (0%)
NeuralRewriter	-38575.78 (9.3%)	-27350.68 (11.72%)
Q-net	-37386.9 (5.93%)	-26031.5 (6.33%)
MCTS+WSPT	-35489.56 (0.55%)	-24724.76 (0.99%)
SOLO	-35434.46 (0.4%)	-24747.38 (1.08%)
SOLO+Prune	**-35280.2 (-0.04%)**	**-24655.42 (0.71%)**

Figure 5: Scheduling results for all problem variants. Each cell includes the average cost on 50 seeds and the fractional improvement of each method compared to CPLEX.

Offline CVRP	20	100
Uniform-Random[UR]	-13.21(107.51%)	-58.84 (230.13%)
Distance[D]	-10.43 (63.65%)	-47.59 (167.38%)
Savings	-6.35 (-1.04%)	**-16.51 (-7.94%)**
Sweep	-8.89 (39.33%)	-28.24 (58.11%)
OR-Tools	-6.42 (0.00%)	-17.96 (0.00%)
NeuralRewriter	-6.95 (8.48%)	-19.45 (8.57%)
Q-Net	-6.84 (6.59%)	-19.27 (7.62%)
MCTS+UR	-7.65 (19.45%)	-46.34 (160.11%)
MCTS+D	-7.15 (12.01%)	-44.00 (147.44%)
SOLO	**-6.21 (-3.18%)**	-17.68 (-1.24%)

Online CVRP	20	100
Uniform-Random[UR]	-12.72 (31.67%)	-52.73 (108.06%)
Distance[D]	-9.75 (0.76%)	-33.65 (32.72%)
Savings	-9.90 (0.51%)	-25.15 (-0.90%)
Sweep	-11.16 (13.73%)	-29.52 (16.16%)
OR-Tools	-9.86 (0.00%)	-25.40 (0.00%)
NeuralRewriter	-10.00 (1.56%)	-25.85 (1.90%)
Q-net	-8.79 (-9.76%)	-26.80 (5.70%)
MCTS+UR	-7.80 (-20.27%)	-28.72 (12.98%)
MCTS+D	-6.78 (-30.84%)	-25.58 (0.78%)
SOLO	**-6.63 (-32.38%)**	**-24.80 (-2.28%)**

Figure 6: Offline and Online CVRP results. Each cell contains the average cost and the fractional improvements over OR-Tools.

CONCLUSION AND FUTURE WORK

- A hybrid Learning-planning scheme for dealing with NP-Hard CO problems
- Size generalization with compact network by virtue of the graph representation
- Refinement of learning approximations with online search.

- Close the loop by integrating the online MCTS experience back into the learning stage.

REFERENCES

1. Silver, D. et. al. 2017. Mastering the game of go without human knowledge. Nature, 550(7676):354–359.
2. Kocsis, L. et. Al. C. 2006. Bandit based monte-carlo planning. In ECML, 282–293. Springer.
3. Zhuwen, L. et. al. 2018. Combinatorial Optimization with Graph Convolutional Networks and Guided Tree Search. In NeurIPS.