Reinforcement Learning, metaheuristics and Combinatorial Optimization Problems

Chen Jingjing, 55766475

June 22, 2020

Abstract

This article includes the introduction and methodologies of Reinforcement Learning(RL) and some heuristic methods to solve Combinatorial Optimization Problems.
Keywords: reinforcement learning, combinatorial optimization problems, heuristic search.

1 Combinatorial Optimization Problems

1.1 Travel Salesman Problem

This part has been written in the report of CS8692.

1.2 Vehicle Routing Problem

This part has been written in the report of CS8692.

1.3 Quadratic Assignment Problem(QAP)

$$\min_{\pi} \quad g(\pi),$$

$$\mathrm{g}(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{\pi_i \pi_j}$$

Where:
- n is the size of the problem (i.e., number of facilities or locations),
- π is a permutation, where π_i is the $i-th$ element in the permutation π.
- a_{ij} and $b_{\pi_i \pi_j}$ are the element of $n \times n$ distance and flow matrices A and B.

QAP is to find a permutation π_i(which represents which facilities are placed at which locations), which minimizes the sum of the distance times the flow between different facilities. Each element a_{ij} of the matrix a represents the distance between location i and location j. The element bij represents the flow between facilities π_i and π_j . When a_{ij} is multiplied by $b_{\pi_i \pi_j}$, the cost of placing facility π_i at location i and facility π_j at location j, is obtained. Thus, by summing all the terms together, the total cost of the whole permutation of location-facility assignments is obtained.

2 metaheuristics

2.1 Guided local search

This part has been written in weekly report before and will update soon.

2.2 GRASP

This part has been written in weekly report before and will update soon.

2.3 Tabu search

Waiting to update.

3 Metaheuristics for Combinatorial Optimization Problems

3.1 GLS for TSP

The features: edges e_{ij}
Penalty: $p_{ij} = 1 + p_{ij}$
cost function: d_{ij}
utility function: $I_{e_{ij}}(tour) * \frac{d_{ij}}{1+p_{ij}}$
new cost function: $d_{ij}+\lambda p_{ij}$
local search methods: 2-Opt, 3-Opt, LK.

3.2 GLS for QAP

3.2.1 PAPER 1: Applying an extended guided local search to the quadratic assignment problem

This article [3] show how an extened Guided Local Search can be applied to QAP. Authors introduced the formulation of QAP and basic counterparts of GLS(selective penalty modifications and the augmented cost function). Here it used the stochastic 2-opt method to find a better solution with iterations. If the iterations is not big enough, maybe it's not possible to swap the local minima. The local search neighbourhood is simply the set of possible permutations resulting from the current permutation with any two of the elements transposed.

After the exchange of units r and s in the permutation π, the change in objective function is

$$\Delta g(\pi, r, s) = 2 \sum_{k=1, k\neq rs}^{n} (a_{rk} - a_{sk}) (b_{\pi_s \pi_k} - b_{\pi_r \pi_k}) \tag{3.1}$$

The feature set is the facility-location assignments. The cost of a particular facility-location assignment(feature) is the sum of the constituent parts of the objective function:

$$\text{cost}\,(i, \pi_i) = \sum_{j=1}^{n} a_{ij} b_{\pi_i \pi_j} \tag{3.2}$$

$\Delta h(\pi, r, s)$ is the change in augmented cost h of permutation π, after the elements i and j have been swapped; p_{i,π_i} is the penalty when the $i - th$ element of permutation π is assigned the value π_i.

$$\Delta h(\pi, r, s) = \Delta g(\pi, r, s) + \lambda \left((p_{r,\pi_s} + p_{s,\pi_r}) - (p_{r,\pi_r} + p_{s,\pi_s}) \right) \tag{3.3}$$

The key point about using GLS to solve QAP are the selection of features for QAP and the augmented cost function. By means of the process of GLS, we can get the pseudocode for GLSQAP.

GLS is a general, penalty-based meta-heuristic method, which sits on top of local search algorithms, to help guide them out of local minima.

4 Reinforcement Learning

4.1 Theory-based articles

4.1.1 Paper 1: Value function based reinforcement learning in changing Markovian environments

Main idea

How will the value function change when the environment changes? This paper shows the change of applying value function based reinforcement learning (RL) methods to the environment which may change over time.

First, the optimal value function Lipschitz continuously depends on the immediate-cost function(instant reward) and the transition probability function(in this paper, the TP-function is the transition probability). And the dependence on the discount-factor is non-Lipschitz but the change of the discount-factor can be transformed to the variation of the cost-function.

for the common case in MDP, the dynamic state transition process is as follows:

$$x_{t+1} = f(x_t, a_t, w_t) \tag{4.1}$$

There are several basic theorems:

Theorem 1 Let M be a discounted MDP and J is an arbitrary value function. The value function of the greedy policy based on J is denoted by J^π. Then, we have

$$\|J^\pi - J^*\|_\infty \leq \frac{2\alpha}{1-\alpha} \|J - J^*\|_\infty \tag{4.2}$$

This theorem shows that if the arbitrary value function closes to the optimal value function $J^\star$, then we can get a more approximate value function J^π based on this value function. Meanwhile, we get a better policy, for example, greedy policy.

Theorem 2 Assume that two discounted MDPs differ only in their transition functions, denoted by p_1 and p_2. Let the corresponding optimal value functions be J_1^* and J_2^*, then

$$\|J_1^* - J_2^*\|_\infty \leq \frac{n\alpha\|g\|_\infty}{(1-\alpha)^2} \|p_1 - p_2\|_\infty \tag{4.3}$$

recall that n is the size of the state space and $\alpha \in [0, 1)$ is the discount rate.

Theorem 3 Assume that two discounted MDPs differ only in the immediate-costs functions, g_1 and g_2. Let the corresponding optimal value functions be J_1^* and J_2^*, then

$$\|J_1^* - J_2^*\|_\infty \leq \frac{1}{1-\alpha} \|g_1 - g_2\|_\infty \tag{4.4}$$

Theorem 4 Assume that two discounted MDPs differ only in the discount factors, denoted by $\alpha_1, \alpha_2 \in [0, 1)$. Let the corresponding optimal value functions be J_1^* and J_2^*, then

$$\|J_1^* - J_2^*\|_\infty \leq \frac{|\alpha_1 - \alpha_2|}{(1-\alpha_1)(1-\alpha_2)} \|g\|_\infty \tag{4.5}$$

These theorems guarantees the analysis of the following content.

Secondly, $(\epsilon, \delta) - MDP$ was introduced, which is a generalization of MDP and $\epsilon - MDP$. Applying these result in $(\epsilon, \delta) - MDP$, in which these functions are allowed to vary over time, as long as the cumulative changes remain bounded in the limit with time t.

Finally, they verified the feasibility of classical RL methods like Q-learning, value iteration and TD learning in the environment may changing over time to time. More precisely, the immediate-cost function and the transition probability function may vary from time to time.

Comments This is an novel perspective for considering the change of the environment for me. The theoretical guarantee may be helpful with my initial idea.

4.2 Articles about ML-based methods to solve combinatorial optimization problem

Related articles have been updated to Github and will update in this article soon.

References

[1] Handbook of Metaheuristics[M]. Springer Science Business Media, 2003.

[2] Abdel-Basset M, Manogaran G, Rashad H, et al. A comprehensive review of quadratic assignment problem: variants, hybrids and applications[J]. Journal of Ambient Intelligence and Humanized Computing, 2018: 1-24.

[3] Mills P, Tsang E, Ford J. Applying an extended guided local search to the quadratic assignment problem[J]. Annals of Operations Research, 2003, 118(1-4): 121-135.

[4] Csáji B C, Monostori L. Value function based reinforcement learning in changing Markovian environments[J]. Journal of Machine Learning Research, 2008, 9(Aug): 1679-1709.