Bandit Algorithms for Neighborhood Selection in Local Search

Stefano Michelini, Renaud De Landtsheer
Combinatorial Algorithmics, CETIC research center, Charleroi
e-mail: {stefano.michelini, renaud.delandtsheer}@cetic.be

Moisés Silva-Muñoz, Alberto Franzin
Université Libre de Bruxelles (ULB)
e-mail: {moises.silva.munoz,alberto.franzin}@ulb.be

Augustin Delecluse
TRAIL, ICTM, UCLouvain
email: augustin.delecluse@uclouvain.be

Keywords: Local search, Bandit algorithms, Reinforcement learning

1 Neighborhood selection

When tackling a combinatorial optimization (CO) problem using local search, choosing the right neighborhood functions is crucial for good performance. This is especially true for methods such as variable neighborhood search, where different neighborhood functions are used sequentially. Common selection policies include *round robin*, that iterates through all the available neighbourhoods, and *hill climbing*, that prioritizes the ones that allow to obtain the best improvement. The *best slope first* policy prioritizes the neighborhoods that exhibit the best improvement in solution quality relative to the elapsed computing time. The latter is one of the default policies in OscaR.cbls, the local search framework for CO used in this study [1].

2 Bandit algorithms for neighbourhood selection

In recent years, the application of machine learning techniques for the acceleration of more traditional techniques to solve CO problems has garnered considerable attention [2]. In particular, reinforcement learning (RL) models the behaviour of an agent interacting with an *environment*. The agent's actions change the state of the environment, and are each associated to a *reward*. The goal is to learn a policy of actions that maximizes the total reward.

One class of such methods is multi-armed bandit algorithms, where each action is a choice among a set of competing alternatives. The agent needs to learn the optimal schedule of a limited amount of actions. Recent examples of hybdrization between CO and bandit algorithms include [3], where an agent dynamically selects the variable to branch on during the exploration of a search tree, or [4], where the choice is among several heuristics for branch-and-bound exploration.

Inspired by [4], we define the environment as the state of the local search at each iteration and the action space as the set of available neighbourhoods. The optimal policy is the schedule of neighborhoods that yields the best possible solution. The policy is learned *online*, that is, during the search, without a preliminary training phase. We define several reward schemes, such as rewards depending on the presence of improvement, or proportional to the improvement in solution quality, with different frequencies of update. We also implement different bandit algorithms, including the popular ϵ-greedy and UCB, as well as custom implementations. We evaluate whether this approach can define a neighborhood schedule that outperforms the default best slope first method.

3 Experiments

We evaluated these algorithms on the pickup and delivery problem with time windows, for which several neighborhoods are defined. In our experiments we used the benchmark set by Li and Lim [5].

Preliminary results show that some of these algorithms, notably UCB and ϵ-greedy, perform at least as well as the best slope first method, while other alternatives exhibit higher variance in solution quality. These results may anyway be improved by exploiting the higher flexibility of these methods, entailed by their parameterization, with respect to the best slope first policy.

Further studies are therefore aimed at enhancing overall performance by following several directions, such as introducing a preliminary learning phase to optimize algorithmic parameters, as well as refining the definition of the reward schemes. The inclusion of different bandit algorithms is also under consideration. Additionally, since the implementation of these methods is modular and readily available for other problems defined in OscaR.cbls, further research may include the evaluation of their performance on different classes of CO problems.

Acknowledgement This work is supported by the Service Public de Wallonie Recherche under grant n°2010235 - ARIAC. The authors are active members of the TRAIL Institute (Trusted AI Labs, Fédération Wallonie-Bruxelles, Belgium).

References

[1] De Landtsheer, R., and Ponsard, C. (2013) *OscaR.cbls: an open source framework for constraint-based local search.* ORBEL 2013.

[2] Bengio, Y., Lodi, A., and Prouvost, A. (2021). *Machine learning for combinatorial optimization: A methodological tour d'horizon.* EJOR.

[3] Chalumeau F. *et al.* (2021) *Seapearl: A constraint programming solver guided by reinforcement learning.* CPAIOR 2021.

[4] Chmiela, A., Gleixner, A., Lichocki, P., Pokutta, S. (2023). *Online Learning for Scheduling MIP Heuristics.* CPAIOR 2023.

[5] Li, H., and Lim, A. (2001). *A metaheuristic for the pickup and delivery problem with time windows.* ICTAI 2001.