Searching k-Optimal Goals for a Orienteering Problem on a Specialized Graph with Budget Constraints

Abhinav Sharma, Advait Deshpande, Yanming Wang, Xinyi Xu

[1]University of Melbourne
Parkville Campus
Victoria, Australia 3010

Abstract

In this paper, we are proposing a novel non-randomized anytime orienteering algorithm for finding k-optimal goals that maximizes reward on a specialized graph with budget constraints. This specialized graph represents a real-world scenario of finding k-most optimal goal states which is analogous to an orienteering problem (OP).

Introduction

Informative Path Planning (IPP) is an exquisite planning problem that aims at finding the most informed path from a pre-defined start state to a goal state that maximises the reward with respect to the budget constraints. On the other hand, Orienteering Problem (OP) is a special case of the IPP problem where reward at different nodes are calculated independently of each other. However, the OP is considered to be NP-hard and mostly solved with heuristic-based search strategies and customized algorithms (Wei and Zheng 2020).

In this research work, we are trying to solve a domain-related orienteering problem which can formalized for a specialized directed weighted graph. This specialized graph maps the problem of finding the most optimal nearest building or nearest floor from a starting building or starting floor such that reward can be maximised within the provided travelling budget constraint. Using our proposed non-randomized algorithm, we are going to find the k-most optimal goals for our defined exploration problem.

Problem Formulation

In this section, we are going to formulate our domain-related optimal building or floor finding problem into a generic orienteering problem (OP).

Let us assume a weighted directed specialized graph $G_s = (V, E)$ for n number of nodes where $v_s \in V$ is the pre-defined start node such that,

$$V = \{v_1, v_2, v_3, ..., v_n\} \tag{1}$$

$$E = \{(v_s, v_i) \setminus (v_s, v_s) \mid \forall\, i \in [1, n]\ \} \tag{2}$$

where v_s is having n out-degree with 0 in-degree i.e connected to every other node in V and $v_i\ \forall\ i \in [1, n] \setminus v_s$ is connected to only v_s with 1 in-degree and 0 out-degree.

Let v_g be the set of k-optimal goal nodes s.t. $v_g \in V$ and $k \leq n$. These goals are attained in the decreasing order of their gained rewards after respecting budget constraints i.e. $v_{g1} > v_{g2} > ... > v_{gk}$. Let r be the set of nodes which we can visit such that $r \subseteq V \setminus v_s$. Let B be the travelling budget which will help us to enable budget constraints. Let O be the generic objectives and F be the generic factors which can be used to tweak the reward function of our problem.

For each r, let $R(r, o, f)$ be the reward function where $R : (r, o, f) \rightarrow \mathbb{R}_0^+ \cup \{\infty\}$ calculates reward based on the provided factor $f \in F$ and objective $o \in O$. Let $I(r) = R(r, o, f)$ be the reward gained by visiting each node in r. Let the cost of traversal be given by $C(r) = C(v_s, v_i^r)$, where v_i is the i^{th} element in r, $\forall i \in [1, |r|]$. Let $L \in \mathbb{R}_0^+ \cup \{\infty\}$ be the constraint limit. Using above notations, the hard-constraint problem can then be defined by the equation 3.

$$\underset{r \subseteq V}{\arg\max}\, I(r) \textit{ subject to } C(r) \leq B \leq L \tag{3}$$

We can relax the above hard-constraint by introducing a hyperparameter δ to formulate a soft-constraint problem as shown in the equation 4.

$$\underset{r \subseteq V}{\arg\max}\, I(r) \textit{ subject to } C(r) \leq B + \delta \leq L \tag{4}$$

where $\delta \in \mathbb{R}_0^+ \cup \{\infty\}$.

The solution to our stated problem is a set of ordered k-optimal goal nodes, such that the reward attained by visiting the node is maximized while the path cost stays within a specified travelling budget B.

Non-randomized Anytime Orienteering

In this section, we propose a novel way of solving the problem formulation shown in the equation 4 which is inspired by the general randomized algorithm for IPP problems (Arora and Scherer 2017).

The algorithm starts with a priority queue and creates r subset s.t. $r \subseteq V \setminus v_s$. Then, for each node in r, path cost $C(r)$ and node reward $I(r)$ is calculated. It is then ensured that the budget constraint is satisfied and selected node is pushed into the priority queue with negative reward as priority. As a result, we can easily pop the queue item with minimum priority k-times to find the k-most optimal goal nodes. This process is described in Algorithm 1.

Algorithm 1: Non-Randomized Anytime Orienteering to find k-optimal goals for a specialized graph

Inputs: $G_s = (V, E), v_s, B, L, k, \delta, o \in O, f \in F$
Output: $v_g = \{v_{g1}, .., v_{gk}\}$ s.t. $v_{g1} > ... > v_{gk}$
queue := new priority queue
$v_g = \emptyset$
$r := r \subseteq V \setminus v_s$
for v_i *in* r **do**
  $I(r) = R(r, o, f)$ `//node reward`
  $C(r) = C(v_s, v_i)$ `//path cost`
  if $C(r) \leq B + \delta \leq L$ **then**
    $priority = -1 * I(r)$
    queue.insert($v_i, priority$)
  end
end
while *not queue.empty()* **do**
  ρ := queue.pop-min() `//best node`
  if $len(v_g) < k$ **then**
    $v_g := v_g \cup \rho$
  end
end

Properties & Limitations

In this section, we will list some of the properties and drawbacks of the algorithm.

Time Complexity. If we assume standard binary heap implementation of the priority queue then the insertion and deletion time complexity is $O(log\ n)$ where n is the size of the input (Atkinson et al. 1986). This can be further optimized by several customisation (Edelkamp, Elmasry, and Katajainen 2017). Hence, the time complexity of our proposed algorithm for best and worst case can be stated as

$$O(n - 1 * log\ n) + O(k * log\ n) \leq O(n\ log\ n) \quad (5)$$

Space Complexity. With priority queue, the entries of the queue are kept sorted and entry with lowest priority is retrieved first. So, if we again assume heap data structure implementation of the priority queue, then the space complexity of storing n elements in the priority queue is $O(n)$ (Atkinson et al. 1986). Hence, the best and worst case space complexity of our proposed algorithm is $O(n)$.

Limitations. Our algorithm relies on the assumption that the graph is a weighted directed graph with 1 central node (0 in-degree and n out-degree) and n isolated nodes connected with only 1 central node. Due to this assumption, this algorithm is efficient and applicable only for such versions of the specialized graph and cannot be extended implicitly to any general weighted directed graph.

Experimental Results

In this section, we are going to show experimental results for a domain-specific orienteering problem solved using our proposed algorithm. In our problem, we need to find the most optimal nearest building within a specific radius (B) that maximises the chances (reward) of either booking a meeting room or using a toilet facility based on supply, demand and other preferences or factors. A specific scenario is shown in Figure 1.

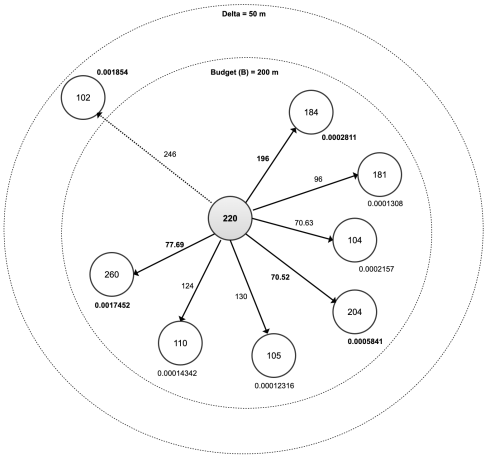

Figure 1: Finding $k = 3$ most optimal nearest building from $v_s = 220$ that maximises the chances (reward) of booking a meeting room within $B = 200$ meters and $\delta = 50$ meters

Using our stated algorithm, we were able to solve above shown scenario with the following results.

Goals	Cost	Reward
$v_{g1} = 260$	77.69	0.0017450
$v_{g2} = 204$	70.52	0.0005841
$v_{g3} = 184$	196	0.0002811

Goals	Cost	Reward
$v_{g1} = 102$	246	0.0018540
$v_{g2} = 260$	77.69	0.0017450
$v_{g3} = 204$	70.52	0.0005841

Table 1: $k = 3$ most optimal nearest buildings with $B = 200$ hard-constraint (left) and $B+\delta = 250$ soft-constraint (right)

Conclusion

In this paper, we formulated a domain-specific problem into a general specialized graph based orienteering problem with budget constraints. We also proposed non-randomized anytime orienteering algorithm which solves the formulated problem with $O(n\ log\ n)$ time complexity. Preliminary results were also presented for a specific scenario to highlight the efficiency and usability of the algorithm.

References

Arora, S.; and Scherer, S. 2017. Randomized algorithm for informative path planning with budget constraints. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 4997–5004.

Atkinson, M. D.; Sack, J.-R.; Santoro, N.; and Strothotte, T. 1986. Min-Max Heaps and Generalized Priority Queues. *Commun. ACM* 29(10): 996–1000. ISSN 0001-0782. doi:10.1145/6617.6621. URL https://doi.org/10.1145/6617.6621.

Edelkamp, S.; Elmasry, A.; and Katajainen, J. 2017. Optimizing Binary Heaps. *Theory of Computing Systems* 61. doi:10.1007/s00224-017-9760-2.

Wei, Y.; and Zheng, R. 2020. Informative Path Planning for Mobile Sensing with Reinforcement Learning. doi:10.1109/INFOCOM41043.2020.9155528.