A Hybrid Firefly and Local Search Algorithms to Solve Orienteering Problems

C Sitompul, R Pramudi and F Setiawan

Department of Industrial Engineering, Parahyangan Catholic University, Bandung, Indonesia

E-mail: carles@unpar.ac.id

Abstract. A satisfied customer is a goal for every tourism agency who put efforts designing a tour route for its customers. The resulting problem often called an orienteering problem attempts to design a route that maximizes customer’s value for all destinations. We propose a firefly algorithm combined with a hybrid of local search algorithms to solve the problem. Our finding however suggests that firefly algorithms cannot outperform Ant Colony (AC) and Genetic Algorithm (GA).

1. Background

For Indonesia, tourism has been one of the biggest contributors to its gross domestic products. In 2018, Indonesia attracted more than 15.8 million foreign tourists, reflecting a 12% growth from the previous year [1]. However, Indonesia still runs behind Thailand, Singapore and Malaysia which forces the Indonesian Government to ramp up its tourism sector. Tour and travel agencies have the potential to support the industry through designing routes that maximize customers’ satisfaction. This routing problem is often called as an orienteering problem [2]. For example, a travel agency provides a tour package in Bali for a day tour visiting places of interest such as Pura Tanah Lot, Pura Uluwatu, Ubud Monkey Forest, Jati Luwih Rice Fields, Bedugul Lake, and Kuta Beach. A customer’s satisfaction can be described using rating systems for those places, e.g. given in TripAdvisor.

The orienteering problem has been discussed for decades starting with Tsiligirides first attempt to solve the problem using a heuristic method [3]. Since its introduction, the problem has evolved to many variations for example Time Dependent Orienteering Problem (TDOP), Orienteering Problem with Time Windows (OPTW), and Team Orienteering Problem (TOP). The orienteering problem can be seen as a combination of two classical combinatorial problems, i.e. Knapsack Problem and Travelling Salesmen Problem [2]. Furthermore, the orienteering problem is categorized as an NP-hard problem [4] where solutions can only be found in a non-polynomial time or solutions cannot be found. This orienteering problem can be formulated as in Vansteenwegen [2] as follows. Suppose $N$ is the number of nodes or destinations, where $p_i$ denotes profit of visiting $i$, and $c_{ij}$ as costs of transportations from $i$ to $j$, $i, j \in \{1 \ldots N\}$. Variable $x_{ij}$ equals to 1 if a route exists from $i$ to $j$ and variable $\mu_i \geq 0$ denotes node $i$ in the route assuming that the route starts at node 1 and ends in node $N$.

Maximize

$$\sum_{i=1}^{N-1} \sum_{j=2}^{N} p_i x_{ij}$$ (1)
Subject to:

$$\sum_{j=2}^{N} x_{ij} = \sum_{i=1}^{N-1} x_{iN} = 1$$ (2)

$$\sum_{i=1}^{N-1} x_{ir} = \sum_{j=2}^{N} x_{rj}, \forall r = 2, \ldots, N - 1$$ (3)

$$\sum_{i=1}^{N-1} \sum_{j=2}^{N} p_{ij} x_{ij} \leq B,$$ (4)

$$2 \leq u_{i} \leq N, \forall i, j = 1, 2, N$$ (5)

$$u_{i} - u_{j} + 1 \leq (n - 1)(n - x_{ij}), \forall r = 2, \ldots, N$$ (6)

$$x_{ij} \in \{0, 1\}, \forall i, j = 2, \ldots, N$$ (7)

2. A Firefly Algorithm

Firefly algorithms are introduced by Xin-She Yang in 2010 [5] inspired by fireflies and their conspicuous use of bioluminescence during twilight to attract mates or prey. Some assumptions used in the firefly algorithms are:

1. Fireflies are unisex
2. Attractiveness is proportional to their brightness.
3. The brightness of a firefly is determined by the objective function.

Based on these three rules, the basic steps can be summarized as pseudo code as in Figure 1.

---

Figure 2. Pseudo code of the firefly algorithm (Yang, 2010)
The firefly algorithm can be implemented to solve the orienteering problem as follows. A firefly’s location is analogous to a solution in orienteering problems, hence finding a better solution can be achieved through a movement of their locations. An exploitation is considered when fireflies are attracted to other fireflies, otherwise a random movement is considered as an exploration. Encoding is a process of translating a solution into a firefly’s location. A vector of firefly’s location has \( (N - 2) \) dimensions where entries are generated using a random number \( U(0,1) \). A decoding is used to translate a vector into solutions, i.e. (1) determining the sequence without constraints, (2) inserting each destination in the sequence into the solution taking into account the constraints, (3) calculating the objective function which is seen as an intensity level. The light intensity \( I(r) \) varies according to the inverse square law and the absorption law following Gaussian form:

\[
I(r) = I_0 e^{-\gamma r^2}
\]

The attractiveness \( \beta \) can be defined by:

\[
\beta = \beta_0 e^{-\gamma r^2}
\]

where \( \beta_0 \) is the attractiveness at \( r = 0 \). The attractiveness can also be approximated as:

\[
\beta = \frac{\beta_0}{1 + \gamma r^2}
\]

The distance between two fireflies \( x_i \) and \( x_j \) can be defined as a Cartesian \( d \)-dimension:

\[
r_{ij} = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2},
\]

where \( x_i \) and \( x_j \) denotes vectors of two fireflies.

Fitness value of each firefly is calculated and then compared one another. A firefly \( i \) is attracted to firefly \( j \) if firefly \( j \) has a higher fitness function. If there exist no other fireflies with higher fitness function, then firefly \( i \) will move randomly. As input parameters, we use the following: randomization parameter \( (\alpha) \), attractiveness \( (\beta) \), light absorption coefficient \( (\gamma) \), number of population and number of iterations.

The movement of a firefly is determined by:

\[
x_1 = x_1 + \beta_0 e^{-\gamma r_0^2(x_i - x_j)} + \alpha \epsilon_i
\]

where \( \epsilon_i \) is random error generated from a uniform distribution.
3. Local Search Algorithms
We also propose some local search algorithms to improve the quality of the fitness function provided by firefly algorithm. The fitness function is improved using inner swap algorithm, outer swap algorithm, insertion algorithm, and 2-opt algorithm. Inner swap algorithm is used to swap two nodes within the route, while an outer swap algorithm exchanges a node in the route from a node outside the route. An insertion algorithm is also used to add a node into existing route expecting that it will improve the solution. Lastly, a 2-opt local search is used to reduce the number of crossovers among routes.

4. Computational Results
We implement the proposed algorithm using JAVA Net Beans IDE 8.2 and use it to solve three benchmark problems as suggested in Tsiligirides [3]. The following parameters are used, i.e. number of iterations 10,000 to avoid failures finding the solution. The number of fireflies is experimented using 3 levels: 15, 25 and 40. The randomization parameter (α) and attractiveness (β) both has three levels: 0.1, 0.5 and 1. We use 0,10 and 50 as light absorption coefficient (γ). Using one factor at a time (OFAT) experimental design, we have the following parameter combination as given in Table 1.

Table 1. Parameter combination

| Combination | n  | α    | β    | γ  |
|-------------|----|------|------|----|
| A           | 10 | 0.1  | 0.1  | 0  |
| B           | 25 | 0.1  | 0.1  | 0  |
| C           | 40 | 0.1  | 0.1  | 0  |
| D           | 10 | 0.5  | 0.1  | 0  |
| E           | 10 | 1.0  | 0.1  | 0  |
| F           | 10 | 0.1  | 0.5  | 0  |
| G           | 10 | 0.1  | 1.0  | 0  |
| H           | 10 | 0.1  | 0.1  | 10 |
| I           | 10 | 0.1  | 0.1  | 50 |

From all three datasets, the firefly algorithm performs worse than any best-known solution averaging, i.e. 5%, 8% and 10% worse in the first, second and third datasets respectively. Even when compared with Ant Colony Algorithm [6] and Genetic Algorithm [7], the firefly algorithm also performs worse. However, when combined with local search, the algorithm performs better achieving almost 60% of best-known solutions for all datasets. When compared with Ant Colony (AC) and Genetic Algorithm (GA), the proposed algorithm differs less than 2% worse in dataset 1 and dataset 3. In dataset 2, the algorithm can reach the same level with both AC and GA algorithms.

5. Conclusion
In general, firefly algorithm in itself cannot outperform Ant Colony (AC) and Genetic Algorithm (GA). However, when combined with local search algorithms the algorithm may reach the same level as those algorithms. We propose that further research should be directed to a more effective way to implement the algorithm within the JAVA programming environment.

ACKNOWLEDGMENTS
We would like to thank Directorate General of Learning and Student Affair, Ministry of Research Technology and Higher Education for giving us the financial support in this publication as implementation of grant program in the postgraduate improvement quality by the contract number (No.036/B4/PPK-SPK/VII/2019).
References

[1] Post J 2019 Indonesia welcomed 15.8 million foreign tourists last year: BPS
[2] Vansteenwegen P, Souffriau W and Van Oudheusden D 2011 The orienteering problem: A survey Eur. J. Oper. Res. 209 1–10
[3] Tsiigirides T 1984 Heuristic methods applied to orienteering J. Oper. Res. Soc. 35 797–809
[4] Golden B L, Levy L and Vohra R 1987 The orienteering problem Nav. Res. Logist. 34 307–18
[5] Yang X-S 2010 Nature-inspired metaheuristic algorithms (Luniver press)
[6] Liang Y-C and Smith A E 2006 An ant colony approach to the orienteering problem J. Chinese Inst. Ind. Eng. 23 403–14
[7] Tasgetiren M F and Smith A E 2000 A genetic algorithm for the orienteering problem Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512) vol 2 (IEEE) pp 910–5