A Hybridized Scheme for Solving Ridesharing Problems Based on Firefly Algorithm and a Variant of PSO Algorithm

Fu-Shiung Hsieh[a]

Department of Computer Science and Information Engineering, Chaoyang University of Technology, 168, Jifeng E. Rd., Wufeng District, Taichung, 413310 Taiwan, R.O.C.

Keywords: Ridesharing, Particle Swarm Optimization, Firefly, Discrete Optimization.

Abstract: Solving constrained discrete optimization problems with discrete decision variables poses a challenge due to complexity. Ridesharing problems are usually formulated as constrained discrete optimization problems. An urgent need is to develop an efficient algorithm for solving ridesharing problems. One potential approach to constrained discrete optimization problems is based on hybridization of different evolutionary algorithms to increase diversity in generating candidates during the evolution processes. The goal of this paper is to provide a proof of concept to study effectiveness of hybridizing the strategies of two evolutionary algorithms, the Firefly algorithm and a variant of PSO algorithm. A hybrid algorithm is proposed based on the above mentioned hybridization mechanism to illustrate that the idea. To verify effectiveness of the hybridization mechanism, several experiments were conducted. The experimental results indicate the hybridization mechanism works effectively and finds the solutions more efficiently than the existing hybrid Firefly-PSO algorithm.

1 INTRODUCTION

A lot of decision problems can be modelled as constrained discrete optimization problems. Matching problems in ridesharing systems (Fiedler et al. 2018) and design/analysis of Cyber-Physical Systems (Hsieh, 2022) fall into the category of constrained discrete optimization problems. Solving constrained discrete optimization problems often poses a challenge due to complexity to search the solution space. Ridesharing problems can be described by optimization problems with discrete decision variables. The decision variables must satisfy several constraints such as capacity constraints, non-negative cost savings and time constraints, etc. Therefore, ridesharing problems are usually formulated as constrained discrete optimization problems. Solving these discrete optimization problems poses several challenges. First, the solution space of discrete optimization problems is typically nonconvex. Classical optimization methods cannot be applied to find the solutions of these non-convex discrete optimization problems. Second, the solution space of discrete optimization problems grows exponentially with the problem size. Finally, due to the excessive constraints in the discrete optimization problems, an effective method need to be applied to efficiently move toward the feasible region and find solutions. It is urgently needed to design an efficient solver for ridesharing problems. This study aims to compare effectiveness of different ways of combining the search strategies of existing evolutionary algorithms. This comparative will be helpful for development of an efficient algorithm.

Classical optimization methods cannot be applied to find the solutions of these non-convex discrete optimization problems effectively. Metaheuristic approaches provide alternative methods to find solutions based on automated selection of heuristic rules. Evolutionary computation is based on metaheuristic approaches to solve these complex non-convex optimization problems. A lot of evolutionary algorithms appeared in past decades. For example, Firefly algorithm (Yang, 2009), (Li et al., 2022) and PSO algorithms (Shami, et al. 2022) were proposed to solve optimization problems. Firefly algorithm has been applied to solve parameter estimation problem

[a] https://orcid.org/0000-0003-0208-9937

Hsieh, F.
A Hybridized Scheme for Solving Ridesharing Problems Based on Firefly Algorithm and a Variant of PSO Algorithm.
DOI: 10.5220/0012761300003690

In *Proceedings of the 26th International Conference on Enterprise Information Systems (ICEIS 2024) - Volume 2*, pages 769-774
ISBN: 978-989-758-692-7; ISSN: 2184-4992

(Sarangi at al., 2018), clustering problem (Alam and Muqeem, 2022) and economic dispatch problem (Sulaiman et al., 2012). PSO algorithm has been applied in Electric Power Systems (AlRashidi and El-Hawary, 2009), Task Scheduling (Valarmathi and Sheela, 2017) survey planning (Seah et al., 2015) and ridesharing systems (Hsieh, 2022), (Hsieh, 2023).

Different evolutionary algorithms can be regarded as different mechanisms to automated selecting heuristic rules to optimize performance. An interesting research issue is study whether combining different evolutionary algorithms can improve performance or convergent speed. However, whether combining different evolutionary algorithms to solve these non-convex discrete optimization problems is effective need further study.

The goal of this paper is to provide a proof of concept to study effectiveness of hybridizing the strategies of two evolutionary algorithms, the Firefly algorithm and a simplified version of PSO algorithm called the Simplified PSO (SPSO) algorithm. A hybrid algorithm is proposed based on the above mentioned hybridization mechanism to illustrate that the idea. To verify effectiveness of the hybridization mechanism, several experiments were conducted. The results indicated the proposed hybridization mechanism based algorithm works effectively and find the solutions more efficiently than the original Firefly or PSO algorithms running alone or the hybrid FA-PSO algorithm (Hsieh, 2024).

The structure of this paper is as follows. In Section II, the instance of discrete optimization problem used to study the effectiveness of hybridization will be introduced. The fitness function and the hybridized algorithm are presented in Section III. We compare the results of PSO algorithm, SPSO algorithm, Firefly-PSO algorithm and Firefly-SPSO algorithm in Section IV. In Section V, we conclude this paper and suggest a future research direction based on the results of this study.

2 THE PROBLEM

As this paper aims to provide a proof of hybridization concept, a problem must be selected for testing the proposed algorithm. As the decision problem of shared mobility systems is a complex constrained discrete optimization problems, it is used as a good candidate for testing the hybridization concept. We will summarize the problem formulation as follows. We will propose a hybridized algorithm for this problem later and study the effects of hybridization through experiments.

Let's consider a system with P passengers and D drivers. We assumed that the drivers and riders send requests to the shared mobility system. To formulate the decision problem of a shared mobility system as a constrained discrete optimization problem, the requests sent by passengers and drivers are represented by b_p^P, $\forall p \in \{1,2,3,....P\}$, and b_{dj}^D, $\forall d \in \{1,...,D\}\ \forall j \in \{1,...,J_d\}$ respectively. The details of b_p^P is defined by b_p^P = $(s_{p1}, s_{p2}, s_{p3}, ..., s_{pK}, f_p)$, where s_{pk} is the requested seats at location k and f_p is the original price of passenger p. The details of b_{dj}^D is defined by b_{dj}^D = $(q_{dj1}, q_{dj2}, q_{dj3}, ..., q_{djK}, o_{dj}, c_{dj})$, where j the request index of driver d, q_{djk} is the seats available to pick up passengers at location k, c_{dj} is the travel cost and o_{dj} is the original travel cost of d in case the driver travels alone.

To formulate the optimization problem, we define x_{dj}, $\forall d \in \{1,...,D\}\ \forall j \in \{1,...,J_d\}$ and y_p, $\forall p \in \{1,2,3,....P\}$ as decision variables for drivers' requests and passengers' requests, respectively. If the request b_{dj}^D is accepted, x_{dj} is 1. Otherwise, x_{dj} is zero. If the request b_p^P is accepted, y_p is 1. Otherwise, y_p is zero. To optimize total cost savings, we define the objective function as

$$F(x,y) = \sum_{p=1}^{P} y_p f_p + \sum_{d=1}^{D} \sum_{j=1}^{J_d} x_{dj}(o_{dj} - c_{dj}).$$

We can describe the supply/demand constraints of the seats by

$$\sum_{d=1}^{D} \sum_{j=1}^{J_d} x_{dj} q_{djk} \geq \sum_{p=1}^{P} y_p s_{pk} \quad \forall \mathrm{k} \in \{1,2,...,\mathrm{K}\} \tag{1}$$

We can describe the nonnegative cost savings constraints by

$$\sum_{p=1}^{P} y_p f_p + \sum_{d=1}^{D} \sum_{j=1}^{J_d} x_{dj} o_{dj} \geq \sum_{d=1}^{D} \sum_{j=1}^{J_d} x_{dj} c_{dj} \tag{2}$$

We can describe the constraints that each driver will have at most one request accepted by

$$\sum_{j=1}^{J_d} x_{dj} \leq 1\ \forall d \in \{1,...,D\} \tag{3}$$

In addition, the constraints that decision variables can only be binary are described by

$$x_{dj} \in \{0,1\}\ \forall d \in \{1,...,D\}\ \forall j \in \{1,...,J_d\} \quad (4)$$

$$y_p \in \{0,1\}\ \forall p \in \{1,...,P\} \quad (5)$$

The shared mobility problem is formulated as:

$$\max_{x,y} F(x,y)$$

$$s.t.\ (1),(2),(3),(4),(5)$$

As the problem stated above is a discrete optimization problem, it is not convex. One approach to solving nonconvex problems is to adopt evolutionary algorithms. We will propose an evolutionary algorithm by combining Firefly algorithm and SPSO algorithm.

3 FITNESS FUNCTION AND SOLUTION ALGORITHM

To develop an evolutionary algorithm to solve an optimization problem, a fitness function must be defined. The purpose of the fitness function is to assess the quality of a solution. Therefore, it must consider both the performance and feasibility of a solution. Performance of a solution is evaluated using the objective function. The fitness function must include the objective function. Feasibility of a solution is evaluated based on violation of constraints. There are several ways to evaluate violation of constraints. We adopt the following function $U(x,y)$ to evaluate violation of constraints. Let S_f be the set of all feasible solutions in a generation.

For the worst solution in the generation, the corresponding objective function value of is $S_{f\min} = \min_{(x,y)\in S_f} F(x,y)$. We define

$$U(x,y) = S_{f\min} + U_1(x,y) + U_2(x,y) + U_3(x,y) + U_4(x,y) + U_5(x,y)$$

$$U_1(x,y) = \left| \sum_{p=1}^{P}\sum_{k=1}^{K} \left(\sum_{d=1}^{D}\sum_{j=1}^{J_d} x_{dj} q_{djk} - y_p s_{pk}\right) \right|$$

$$U_2(x,y) = \left| \sum_{d=1}^{D} \left(1 - \sum_{j=1}^{J_d} x_{dj}\right) \right|$$

$$U_3(x,y) = \min\left(\sum_{p=1}^{P} y_p f_p + \sum_{d=1}^{D}\sum_{j=1}^{J_d} x_{dj} o_{dj} - \sum_{d=1}^{D}\sum_{j=1}^{J_d} x_{dj} c_{dj}, 0.0\right).$$

Based on the above function, we define the fitnesss function $F_1(x,y)$ as follows:

$$F_1(x,y) = \begin{cases} F(x,y) \ if \ (x,y) \ satisfies \ constra\,int\,s \ (1)\sim(6) \\ U(x,y) \ otherwise \end{cases}$$

.

As the algorithm developed in this paper is through combining the strategies to update positions in Firefly algorithm and the SPSO algorithm, the mechanisms to update positions in the Firefly algorithm and the SPSO algorithm are introduced first.

The Firefly algorithm works based on several parameters, including distance between firefly i and firefly j, r_{ij}, light absorption coefficient, γ, and attractiveness for firefly i and firefly j, $\beta_0 e^{-\gamma r_{ij}^2}$, ε_{in}^t: a random value in [0, 1] and a constant α in [0, 1],

The Firefly algorithm updates the positions of firefly i according to a better firefly j as follows.

$$vx_{in} \leftarrow x_{in} + \beta_0 e^{-\gamma r_{ij}^2}(x_{jn} - x_{in}) + \alpha\varepsilon_{in}^t$$

$$vy_{in} = y_{in} + \beta_0 e^{-\gamma r_{ij}^2}(y_{jn} - y_{in}) + \alpha\varepsilon_{in}^t$$

The SPSO algorithm updates the positions of firefly i according to firefly j as follows.

The SPSO algorithm generates random variables r_1 with uniform distribution $U(0,1)$ and updates positions of individuals as follows.

$$vx_{in} \leftarrow vx_{in} + c_1 r_1 (Px_{in} - x_{in})$$

$$vy_{in} \leftarrow vy_{in} + c_1 r_1 (Py_{in} - y_{in})$$

To describe the proposed algorithm, we use $t_{\max}$ to denote total iterations and M to denote the population size. The solution found by an individual is represented by Z_i =(x_i , y_i), where $i \in \{1,2,...,I\}$ and x_i and y_i are the vectors for decision variables x and y, respectively. Let N denote the total dimension of x and y. The maximal value of each element in Z_i is $V_{\max}$.

The algorithm proposed is to combine the logic of the standard Firefly algorithm to update positions of firefly i according to a better firefly j with the logic of the standard SPSO algorithm to update positions of firefly i in case firefly j is worse. In short, the pseudo code of the proposed algorithm is as follows.

To describe the proposed algorithm, we use $t_{\max}$ to denote total iterations and M to denote the population size. The solution found by an individual is represented by Z_m =(x_m , y_m), where $m \in \{1,2,...,M\}$ and x_m and y_m are the vectors for decision variables x and y , respectively. Let N denote the total dimension of x and y . The maximal value of each element in Z_m is $V_{\max}$. Let Z_g denote the global best.

The algorithm proposed is to combine the logic of the standard Firefly algorithm to update positions of firefly i according to a better firefly j with the logic of the standard SPSO algorithm to update positions of firefly i in case firefly j is worse. In short, the pseudo code of the proposed algorithm is as follows.

FA-SPSO Algorithm.

Step 1:
$t \leftarrow 0$
Set γ , β_0 and α
For each $i \in \{1,2,...,I\}$
  Generate Z_i
End For
Step 2:
While (generation $t < t_{\max}$)
{
For $i \in \{1,2,...,I\}$
  $\bar{Z}_i = T(Z_i)$
  $F_1(\bar{Z}_i)$
End For
For $i \in \{1,2,...,I\}$
  For $j \in \{1,2,..., I\}$
    If $(F_1(\bar{Z}_i) < F_1(\bar{Z}_j))$
      For $n \in \{1,2,..., N\}$
        $vx_{in} \leftarrow x_{in} + \beta_0 e^{-\gamma r_{ij}^2}(x_{jn} - x_{in}) + \alpha\varepsilon_{in}^t$
        $vy_{in} = y_{in} + \beta_0 e^{-\gamma r_{ij}^2}(y_{jn} - y_{in}) + \alpha\varepsilon_{in}^t$
      End For
    Else
      For $n \in \{1,2,..., N\}$
        Generate random variables r_1 with uniform distribution $U(0,1)$
        $vx_{in} \leftarrow vx_{in} + c_1 r_1 (Px_{in} - x_{in})$
        $vy_{in} \leftarrow vy_{in} + c_1 r_1 (Py_{in} - y_{in})$
      End For
    End If
  End For
For $n \in \{1,2,..., N\}$
  If $vx_{in} > V_{\max}$
    $vx_{in} \leftarrow V_{\max}$
  End If
  If $vx_{in} < -V_{\max}$
    $vx_{in} \leftarrow -V_{\max}$
  End If
  If $vy_{in} > V_{\max}$
    $vy_{in} \leftarrow V_{\max}$
  End If
  If $vy_{in} < -V_{\max}$
    $vy_{in} \leftarrow -V_{\max}$
  End If
Generate $rsid$ from $U(0,1)$, where $U(0,1)$ denotes uniform distribution

$$x_{in} = \begin{cases} 1 \; rsid < T(vx_{in}) \\ 0 \; otherwise \end{cases}, \text{ where } T(x) = \frac{e^{2|x|} - 1}{e^{2|x|} + 1}$$

Generate $rsid$ from $U(0,1)$, where $U(0,1)$ denotes uniform distribution

$$y_{in} = \begin{cases} 1 \; rsid < T(vx_{in}) \\ 0 \; otherwise \end{cases}, \text{ where } T(x) = \frac{e^{2|x|} - 1}{e^{2|x|} + 1}$$

  End For
$Z_i = (x_i, y_i)$
End For
  Update the global best Z_g
    $t \leftarrow t + 1$
}

4 RESULTS

To study effectiveness of the hybridization mechanism adopted in this paper, several test cases were used to perform experiments. The experiments include solving the test cases by PSO, Firefly, SPSO, FA-PSO and FA-SPSO algorithms. The parameters used in the experiments are summarized in this section. The results of experiments will be compared based on the fitness value and generations for finding solutions.

For each algorithm, the population size, M, is set to 30. The maximum generation for each simulation run is t_{max} =10000. The maximal value V_{max} for each element in the solution vector is 4.

For PSO, the parameters ω =0.4, c_1 =0.4, c_2 =0.6 are used. For SPSO, the parameters ω =0.4, c_1 =0.4 are used. For the Firefly, the parameters β_0 =1.0, γ =0.2, α =0.2 are used.

For the FA-PSO, the parameters ω =0.4, c_1 =0.4, c_2 =0.6, β_0 =1.0, γ =0.2, α =0.2 are used.

For the FA-SPSO, the parameters ω =0.4, c_1 =0.4, β_0 =1.0, γ =0.2, α =0.2 are used.

We applied each algorithm to each test case 10 times, record the results and find the average fitness value and generations. The results are listed in Table 1 and Table 2. Table 1 shows the average fitness function values for all algorithms tested. Table 2 shows the average generation values required for all algorithms and all test cases. The results show that each test case can be solved by each algorithm and the average fitness value found by each algorithm is the same. But the average generations needed for different algorithms are largely different. The average generations needed for FA-PSO algorithm and FA-SPSO algorithm are smaller than those of FA and PSO algorithm. Moreover, the average generations needed for FA-SPSO algorithm are smaller than that of FA-PSO algorithm for most test cases in the experiments.

Table 1: Average Fitness Function Values (NP =30).

Case	D	P	PSO	FA	SPSO	FA-PSO	FA-SPSO
1	1	4	8.495	8.495	8.495	8.495	8.495
2	3	10	44.7	44.7	44.7	44.7	44.7
3	3	10	32.998	32.998	32.998	32.998	32.998
4	5	11	70.91	70.91	70.91	70.91	70.91
5	5	12	41.715	41.715	41.715	41.715	41.715

Table 2: Average number of generations (NP =30).

Case	D	P	PSO	FA	SPSO	FA-PSO	FA-SPSO
1	1	4	2.3	2	2	2	2
2	3	10	87	99.5	98	6.3	10.7
3	3	10	80.2	100.5	153.5	9.8	7
4	5	11	418.6	529.9	1045.7	133.6	33.4
5	5	12	654.8	1291.4	1481.8	154.9	70.3

Figure 1 shows the convergence speed of different algorithms for one of the case. It indicates that FA-SPSO algorithm enjoys faster convergence speed.

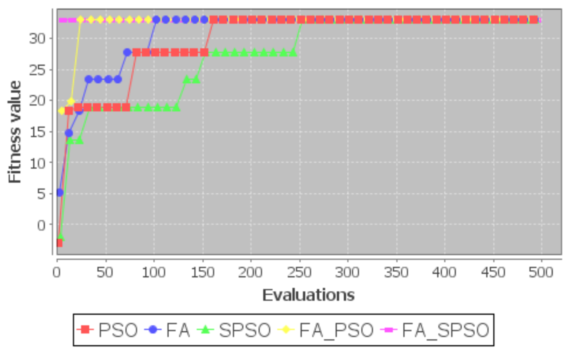

Figure 1: Convergence speed of different algorithms for an example.

5 CONCLUSIONS

Due to computational complexity, constrained discrete optimization problems requires efficient problem solvers to find solutions. Evolutionary computation approach is a practical approach for solving constrained discrete optimization problems. How to improve computational efficiency of evolutionary computation approach in solving constrained discrete optimization problems is an important issue. Intuitively, combining the strategies of two evolutionary computation approaches may be helpful for increasing the diversity of the candidate solutions generated in the evolution processes and may improve either performance or convergence speed. Motivated by this issue, we propose a scheme to combine the strategies of two evolutionary computation approaches. Two evolutionary algorithms were selected to verify our scheme. We study the approach to improve efficiency of evolutionary computation approach through hybridization of Firefly algorithm and SPSO algorithm. We proposed an algorithm by hybridizing Firefly algorithm and SPSO algorithm and study effectiveness of the proposed method. Several test cases were used to test the performance and convergence rates of different algorithms for these test cases. For performance, the results showed that all the algorithms tested can find the same solutions. That is, all of these algorithms perform equally well. The results indicated the algorithm based on hybridization of Firefly algorithm and SPSO algorithm improve the convergence rate. The results of this study sparks an interesting future research direction to study whether the hybridization mechanism can work effectively for other evolutionary computation approaches and problems. For example, effectiveness of hybridization of Firefly algorithm with other meta-heuristic approaches such as the ones proposed in (Hsieh, 2022) and (Hsieh, 2024) is one interesting future research direction.

ACKNOWLEDGEMENTS

This paper was supported in part by National Science and Technology Council, Taiwan, under Grant NSTC-111-2410-H-324-003.

REFERENCES

Alam, A. and Muqeem, M. (2022) Automatic Clustering for Selection of Optimal Number of Clusters by K-Means Integrated with Enhanced Firefly Algorithms, 2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS), Tashkent, Uzbekistan, pp. 343-347.

AlRashidi, M. R. and El-Hawary, M. E. (2009) A Survey of Particle Swarm Optimization Applications in Electric Power Systems, IEEE Transactions on Evolutionary Computation, vol. 13, no. 4, pp. 913-918.

Fiedler, D., Čertický, M. Alonso-Mora, J. and Čáp, M. (2018). The Impact of Ridesharing in Mobility-on-Demand Systems: Simulation Case Study in Prague, 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018, pp. 1173-1178.

Hsieh, F.S. (2022) A Theoretical Foundation for Context-Aware Cyber-Physical Production Systems Applied Sciences 12, no. 10: 5129.

Hsieh, F.S. (2022) An Efficient Method to Assess Resilience and Robustness Properties of a Class of Cyber-Physical Production Systems, Symmetry, vol. 14, no. 11: 2327.

Hsieh, F.S. (2023) Improving Acceptability of Cost Savings Allocation in Rides-haring Systems based on Analysis of Proportional Methods, Systems, vol. 11, no. 4, pp. 187-218.

Hsieh, F.S. (2022) "Development and Comparison of Ten Differential-Evolution and Particle Swarm-Optimization Based Algorithms for Discount-Guaranteed Ridesharing Systems" Applied Sciences, vol.12, no. 19: 9544.

Hsieh, F.S. (2024) Comparison of a Hybrid Firefly–Particle Swarm Optimization Algorithm with Six Hybrid Firefly–Differential Evolution Algorithms and an Effective Cost-Saving Allocation Method for Ridesharing Recommendation Systems" Electronics 13, no. 2: 324.

Hsieh, F.S. (2024) A Self-Adaptive Meta-Heuristic Algorithm Based on Success Rate and Differential Evolution for Improving the Performance of Ridesharing Systems with a Discount Guarantee, Applied Sciences, vol.17, no. 1: 9.

Hsieh, F.S. (2022) Trust-Based Recommendation for Shared Mobility Systems Based on a Discrete Self-Adaptive Neighborhood Search Differential Evolution Algorithm" Electronics 11, no. 5: 776.

Li, J., X. Wei, B. Li, Z. Zeng (2022) A survey on firefly algorithms, Neurocomputing, vol. 500, pp. 662-678.

Sarangi, S. K., R. Panda and A. Sarangi (2018) Design of adaptive IIR filter with modified firefly algorithm for parameter estimation, 2018 Technologies for Smart-City Energy Security and Power (ICSESP), Bhubaneswar, India, 2018, pp. 1-5.

Seah, M. S., W. L. Tung and T. Banks (2015) A Novel Discrete Particle Swarm Optimization approach to large-scale survey planning, 2015 11th International Conference on Natural Computation (ICNC), Zhangjiajie, China, pp. 261-268.

Shami, T.M., A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh and S. Mirjalili (2022) Particle Swarm Optimization: A Comprehensive Survey, IEEE Access, vol. 10, pp. 10031-10061.

Sulaiman, M. H., M. W. Mustafa, Z. N. Zakaria, O. Aliman and S. R. Abdul Rahim (2012) Firefly Algorithm technique for solving Economic Dispatch problem, 2012 IEEE International Power Engineering and Optimization Conference Melaka, Malaysia, Melaka, Malaysia, pp. 90-95.

Yang, X.S (2009) Firefly algorithms for multimodal optimization, CONFERENCE 2009, LNCS, 5792, pp. 169–178.

Valarmathi, R. and T. Sheela (2017) A comprehensive survey on Task Scheduling for parallel workloads based on Particle Swarm optimisation under Cloud environment, 2017 2nd International Conference on Computing and Communications Technologies (ICCCT), Chennai, India, pp. 81-86.