Modelling Finite State Automata to Find Shortest Path

Ni Wayan Parwati Septiani(✉), Purwanti, Syamsiah, Nurfidah, and Mei Lestari

Teknik Informatika Department, Universitas Indraprasta PGRI, South Jakarta, Indonesia
wayan.parwati@gmail.com

Abstract. In this paper, we present a finite state automata (FSA) algorithm to find the shortest path from the origin to the destination. In this study, we chose Margonda as the initial state and UNINDRA Campus A and Campus B as the final states to accept input both from using public transportation or private vehicles. The FSA algorithm finds the shortest path based on the transition function, which can change from one state to another in response to some inputs, so that it can help provide path options to minimize travel time and cost. Besides, the FSA algorithm helps assist the testing process errors that occur in the process of finding the shortest path because the stages of finding the shortest path can be described simply using the FSA algorithm. The result shows that the proposed FSA algorithm has the ability to find the shortest path and can consider to be a solution of single source shortest path problem.

Keywords: Shortest Path · Finite State Automata · Algorithm · graph

1 Introduction

Jakarta is the capital and the largest city in Indonesia. The increasing number of vehicle and people's mobility has become a common issue encountered by large cities like Jakarta. This issue of high mobility should be addressed by the availability of good transportation modes and numerous road infrastructure. Generally, traffic congestion grows as the mobility of transport users grows especially during rush hours. Therefore, we need a solution to this issue. To reach a place faster, we will look for the shortest route between the origin and the destination. This shortest route is also considered during periods of heavy traffic.

Universitas Indraprasta (UNINDRA) PGRI is one of the private universities located in Jakarta. UNINDRA has two campuses; Campus A is located in Jalan Nangka, South Jakarta, and Campus B is located in Jalan Raya Tengah, East Jakarta. During hectic hours, both lecturers and students experienced this heavy traffic. Especially for lecturers that have many classes on the same day at different campuses, they have to be in class on time. Hence, they have to determine the shortest route to get to the destination campus.

Using existing methods or techniques, the shortest path can be determined by calculating the least distance that must be traversed by the lecturer. To determine the best route, one must be aware of the distance between the origin and destination as well as

J. Handhika et al. (Eds.): ICETECH 2022, ASSEHR 745, pp. 417–424, 2023.
https://doi.org/10.2991/978-2-38476-056-5_43

the road's condition. Therefore, the shortest route from the origin to the destination is determined. However, this strategy is also ineffective due to the abundance of alternative paths. The purpose of locating the shortest route is to reduce travel time and expenses.

However this is not always helpful because there are so many alternative routes. The purpose of determining the shortest route is to summarize the journey and save travel costs from the place of origin to the destination.

The shortest path problem still one of the hottest study areas [1]. It is also possibly the most fundamental and important of all combinatorial optimization problems [2]. Moreover, shortest path analysis is a major analytical component of numerous quantitative transportation and communication models [3, 4].

There are several types of shortest route or shortest path problem: (1) The single pair shortest path problem is finding a path between a single pair of vertices [5], (2) Single source shortest path problem is by pinpointing a vertex to be a source vertex then finding the shortest path from pointed source to all other vertices [6, 7], (3) Single destination shortest path problem is the opposite of single source shortest path, which is finding the shortest path from all vertices to one pointed destination vertex [7], and last (4) all pair shortest path problem is finding the shortest path between all vertices [8].

There are many researches in shortest path problem, [9–15]. Many algorithms are applied to overcome the shortest path problem. A* and Ant Colony Algorithm (ACO) is a common algorithm used to solve Single Source Shortest Path (SSSP) problem [16]. The single shortest path search is a well-known process in retrieving information from graph and it has been proven in practical uses such as telecommunication, transportation, and road network. ACO algorithms often applied to routing in communication network [9, 17, 18], and vehicle routing problem [19–22]. Another hybrid algorithm [23] Bellman-Ford and Dijkstra algorithms is proposed to improve the running time bound of Bellman-Ford for graphs with a sparse distribution of negative cost edges.

A Finite State Machine, commonly referred to as a Finite Automaton, is a standard model used in the mathematical foundation of computer science, for e.g. in the formal specification of programming languages [24]. A Finite State Machine is a computing device whose input is a string and whose output is either accept or reject [25]. FSM or FSA is a quintuple consisting of (Σ, S, S_0, δ, F) [24, 25].

Σ is the input alphabet (a finite non-empty set of symbols).

Q is a finite non-empty set of states.

S_0 is an initial state, an element of S.

δ is the state transition function: δ: S $\times$ Σ $\rightarrow$ S.

F is the set of final states, a (possibly empty) subset of S.

Experts and Scholars have conducted much research on FSA. The theory of automata has been applied to game design [26, 27], it has also been applied to designing a vending machine or selling machine [28, 29]. The concept of Finite State Automata was applied to recognize and then capture the pattern in the process of ticket selling machines [28].

This study attempts to model finite state automata in order to calculate the shortest path, in this case from Margonda to UNINDRA Campus A and Campus B.

2 Methods

The proposed model of FSA algorithm is to find the shortest path. The first step was determine the initial state {q0}, next, street names were defined as a state {q0, q1, q2, q3, q4, q5, q6, q7, q8, q9, q10, qf1, qf2}. The next step is define a set of inputs {a, b}. Next, define set of final states. Since there are 2 final destinations, we label them as qf1 and qf2. Then we described the transition function δ.

An NDFA non-deterministic finite automata is represented by a digraph named "state diagram". This digraph is made up of vertices that represent states, arcs that are labeled with input alphabets, an empty single arc for the first state, and double circles for the last state.

3 Results and Discussion

The aim of this study is to show the implementation of Finite State Automata theory in finding the shortest path for students or lecturers that live in Depok particularly around Margonda Raya Street, to get to UNINDRA Campus A and Unindra Campus B. There are many studies conducted to find the shortest path using graph theory. However, this paper presented a model of FSA to find the shortest path from Margonda to campus UNINDRA.

As shown in Fig. 1, we designed a state transition diagram based on FSA theory.

Formal definition of state diagram in figure one are as follows:

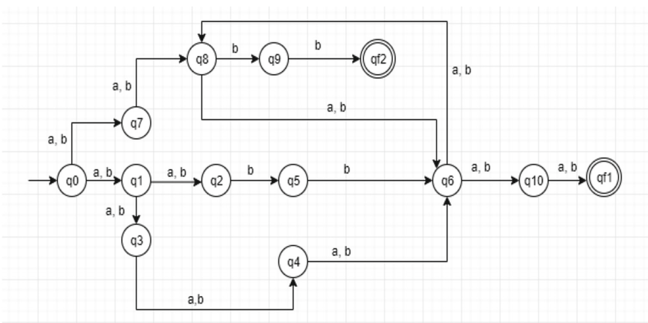

Fig. 1. State Transition Diagram

Q	=	{q0, q1, q2, q3, q4, q5, q6, q7, q8, q9, q10, qf1, qf2}
Σ	=	{a, b}
S_0	=	q0
F	=	{qf1, qf2}

Table 1 and Table 2 provide inputs description and states description.

Table 3 shows the transition of each state. Given the initial state q0 (Margonda) for the input both "a" and b, both using public transport and using private vehicle. The next state will be q7, q1 or q3.

Table 4 shows state transition from initial state to final state campus A and campus B.

There are four possible routes from q0 to final state Campus A (qf2) and fore possible route from q0 to Campus B (qf1), as follows:

1. Start from state q0 (Margonda) to state q7 (Lenteng Agung), next state q8 (Simatupang), next state q9 (Jalan Nangka), last final state qf2 (Campus A)

Table 1. Set of input

Input	Description
A	Initial for travel using public transport
B	Initial for travel using private vehicle

Table 2. State name and its description

State	Description
q0	Margonda
q1	Kelapa dua
q2	R.A. Fadilah
q3	PAL
q4	Pasar Rebo
q5	Kesehatan
q6	Caglak
q7	Lenteng Agung
q8	Simatupang
q9	Jl. Nangka
q10	Raya Tengah
qf1	Campus B
qf2	Campus A

Table 3. Transition Table

States	a	b
q0	q7,q1,q3	q7,q1,q3
q1	q2,q3	q2,q3
q2	-	q5
q3	q4	q4
q4	q6	q6
q5	-	q6
q6	q8,q10	q8,q10
q7	q8	q8
q8	q6,q9	q6,q9
q9	-	qf2
q10	qf1	qf1
qf1	-	-
qf2	-	-

Table 4. State transition to final state

Initial state (S_0)	Next state	Final State	Total state has been passed
q0	q7, q8, q9	qf2	3
q0	q1,q2,q5,q6,q8,q9	qf2	6
q0	q1,q3,q4,q6,q8,q9	qf2	5
q0	q7,q8,q6,q10	qf1	4
q0	q1,q2,q5,q6,q10	qf1	5
q0	q1,q3,q4,q6,q10	qf1	5

2. Start from state q0 (Margonda) to state q1 (Kelapa dua), next heading to state q2 (RA Fadilah), next to state q5 (Kesehatan), Next state q6 (Caglak), next state q8 Simatupang, next state q9 (Jl. Nangka) ends at state qf2 (Campus A)
3. Start from state q0 (Margonda) to state q1 (Kelapa dua) next to q4 (Pasar Rebo) to state q6 (Caglak) to state q8 (Simatupang) to state q9 (Jl. Nangka) ends at state qf2 (Campus A)
4. Start from state q0 (Margonda) to state q7 (Lenteng Agung) next to q8 (Simatupang) next to q6 (Caglak) next to q10 (Raya tengah) and last final state qf1 (Campus B)

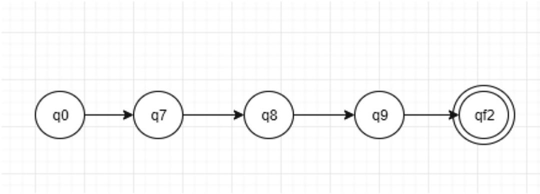

Fig. 2. Minimum State Transition Diagram From q0 to qf2

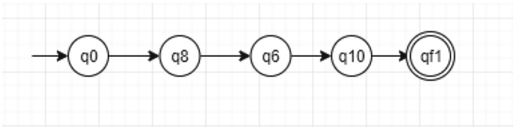

Fig. 3. Minimum State Transition Diagram From q0 to qf1

5. Start from state q0 (Margonda) to state q1 (Kelapa dua) next to q2 (R.A. Fadilah) next to q5 (Kesehatan) next to q6 (Caglak) next to q10 (Raya Tengah) next to final state qf1 (Campus B)
6. Start from state q0 (Margonda) to state q1 (Kelapa dua) next to q3 (Pal) next to q4 (Pasar Rebo) next to q6 (Caglak) next to q10 (Raya Tengah) and to Final state (Campus B)

From the above explanation, shortest path from Margonda to Campus A and Campus B was determined and shown in Fig. 2 and Fig. 3. The transition diagram of minimum state from q0 to qf2 is q0-q7-q8-q9-qf2 (Fig. 2), and the transition diagram from q0 to qf1 is q1-q8-q6-q10-qf1 (Fig. 3).

4 Conclusion

In this study, we used automata theory to find the shortest path. In our methodology, we described an NDFA to find the shortest path from Margonda to destinations at both UNINDRA Campus A and Campus B. The NDFA concept is very helpful and is thought to be a way to solve the single-source shortest path problem.

Acknowledgments. The authors greatly acknowledge UNINDRA PGRI. Special appreciation to reviewers for useful guidance.

Authors' Contributions. Purwanti, Syamsiah, presented the idea, Mei Lestari and Nurfidah verified the analytical methods. Ni Wayan Parwati Septiani wrote the manuscript with input from all authors.

References

1. X. Z. Wang, "The Comparison of Three Algorithms in Shortest Path Issue," in *Journal of Physics: Conference Series*, 2018, vol. 1087, no. 2. doi: https://doi.org/10.1088/1742-6596/1087/2/022011.
2. G. Gallo and S. Pallottino, "A new algorithm to find the shortest paths between all pairs of nodes," *Discret. Appl. Math.*, vol. 4, no. 1, 1982, doi: https://doi.org/10.1016/0166-218X(82)90031-2.
3. B. Yaged, "Minimum cost routing for static network models," *Networks*, vol. 1, no. 2, 1971, doi: https://doi.org/10.1002/net.3230010205.
4. S. Nguyen, "A Unified Approach to Equilibrium Methods for Traffic Assignment," 1976. doi: https://doi.org/10.1007/978-3-642-48123-9_8.
5. S. Geol Baek, S. Lee, and Y. Ik Eom, "Efficient single-pair all-shortest-path query processing for massive dynamic networks," *Inf. Sci. (Ny).*, vol. 546, 2021, doi: https://doi.org/10.1016/j.ins.2020.08.111.
6. H. Arslan and M. Manguoğlu, "A hybrid single-source shortest path algorithm," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 27, no. 4, 2019, doi: https://doi.org/10.3906/elk-1901-23.
7. L. E. Schäfer, T. Dietz, N. Fröhlich, S. Ruzika, and J. R. Figueira, "Shortest paths with ordinal weights," *Eur. J. Oper. Res.*, vol. 280, no. 3, 2020, doi: https://doi.org/10.1016/j.ejor.2019.08.008.
8. A. Brodnik and M. Grgurovič, "Solving all-pairs shortest path by single-source computations: Theory and practice," *Discret. Appl. Math.*, vol. 231, 2017, doi: https://doi.org/10.1016/j.dam.2017.03.008.
9. Kwang Mong Sim and Weng Hong Sun, "Ant colony optimization for routing and load-balancing: survey and new directions," *IEEE Trans. Syst. Man, Cybern. - Part A Syst. Humans*, vol. 33, no. 5, pp. 560–572, 2003, doi: https://doi.org/10.1109/TSMCA.2003.817391.
10. R. Claes and T. Holvoet, "Ant Colony Optimization applied to route planning using link travel time predictions," in *IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum*, 2011, pp. 358–365. doi: https://doi.org/10.1109/IPDPS.2011.173.
11. X. Zhang, Y. Chen, and T. Li, "Optimization of logistics route based on Dijkstra," in *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS*, 2015, vol. 2015-Novem, pp. 313–316. doi: https://doi.org/10.1109/ICSESS.2015.7339063.
12. T. Bhardwaj and S. C. Sharma, "Internet of things: Route search optimization applying ant colony algorithm and theory of computation," in *Advances in Intelligent Systems and Computing*, 2015, vol. 335. doi: https://doi.org/10.1007/978-81-322-2217-0_25.
13. Y. Huang, Q. Yi, and M. Shi, "An Improved Dijkstra Shortest Path Algorithm," *2nd Int. Conf. Comput. Sci. Electron. Eng.*, no. Iccsee, pp. 226–229, 2013, [Online]. Available: http://www.atlantis-press.com/php/download_paper.php?id=4490
14. I. K. Laga Dwi Pandika, B. Irawan, and C. Setianingsih, "Apllication of Optimization Heavy Traffic Path with Floyd-Warshall Algorithm," 2018. doi: https://doi.org/10.1109/ICCEREC.2018.8712110.
15. S. Sanan, L. Jain, and B. Kappor, "Shortest Path Algorithm," *Int. J. Appl. or Innov. Eng. Manag.*, vol. 2, no. 7, pp. 316–320, 2013, [Online]. Available: http://rebustechnologies.com/shortest-path-algorithm-comparison/
16. S. Alani, A. Baseel, M. M. Hamdi, and S. A. Rashid, "A hybrid technique for single-source shortest path-based on a* Algorithm and ant colony optimization," *IAES Int. J. Artif. Intell.*, vol. 9, no. 2, 2020, doi: https://doi.org/10.11591/ijai.v9.i2.pp256-263.
17. G. Di Caro and M. Dorigo, "AntNet: Distributed stigmergetic control for communications networks," *J. Artif. Intell. Res.*, vol. 9, 1998, doi: https://doi.org/10.1613/jair.530.

18. G. Li and Z. Wu, "Ant colony optimization task scheduling algorithm for SWIM based on load balancing," *Futur. Internet*, vol. 11, no. 4, 2019, doi: https://doi.org/10.3390/fi11040090.
19. O. S. da Silva Júnior, J. E. Leal, and M. Reimann, "A multiple ant colony system with random variable neighborhood descent for the dynamic vehicle routing problem with time windows," *Soft Comput.*, vol. 25, no. 4, 2021, doi: https://doi.org/10.1007/s00500-020-05350-4.
20. J. E. Bell and P. R. McMullen, "Ant colony optimization techniques for the vehicle routing problem," *Adv. Eng. Informatics*, vol. 18, no. 1, pp. 41–48, 2004, doi: https://doi.org/10.1016/j.aei.2004.07.001.
21. L. Gambardella, É. Taillard, and G. Agazzi, "MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows," *New Ideas Optim.*, no. IDSIA-06–99, 1999.
22. Z. Xiao and W. Jiang-qing, "Hybrid Ant Algorithm and Applications for Vehicle Routing Problem," *Phys. Procedia*, vol. 25, pp. 1892–1899, 2012, doi: https://doi.org/10.1016/j.phpro.2012.03.327.
23. Y. Dinitz and R. Itzhak, "Hybrid Bellman–Ford–Dijkstra algorithm," *J. Discret. Algorithms*, vol. 42, 2017, doi: https://doi.org/10.1016/j.jda.2017.01.001.
24. F. Wagner, R. Schmuki, T. Wagner, and P. Wolstenholme, *Modeling software with finite state machines: A practical approach.* 2006. doi: https://doi.org/10.1201/9781420013641.
25. E. Rich, "Automata, Computability and Complexity - Theory and Applications," 2019.
26. K. Fathoni, R. Y. Hakkun, and H. A. T. Nurhadi, "Finite State Machines for Building Believable Non-Playable Character in the Game of Khalid ibn Al-Walid," in *Journal of Physics: Conference Series*, 2020, vol. 1577, no. 1. doi: https://doi.org/10.1088/1742-6596/1577/1/012018.
27. Y. Yusoff, A. Nazmi, M. Izzat, M. S. Irwan, M. Zulfahmi, and R. Sallehuddin, "Hangman–Hangaroo Game Design Using Automata Theory," *Int. J. Innov. Comput.*, vol. 11, no. 1, 2021, doi: https://doi.org/10.11113/ijic.v11n1.275.
28. F. J. Kaunang and J. Waworundeng, "Implementation of Finite State Automata in an Amusement Park Automatic Ticket Selling Machine," *Abstr. Proc. Int. Sch. Conf.*, vol. 7, no. 1, 2019, doi: https://doi.org/10.35974/isc.v7i1.1979.
29. F. Ismaya, O. Kurniawan, H. B. Novitasari, J. L. Putra, and W. Gata, "Deterministic Finite State Automatic Smartphone Sales at Handphone Store Using Vending Machine," *J. INFORMATICS Telecommun. Eng.*, vol. 5, no. 2, 2022, doi: https://doi.org/10.31289/jite.v5i2.5489.