

Oadrp Having Issues with Due Date and Limitations with Information

Krishanveer Singh^a

^aAssistant Professor, Institute of Business Management, GLA University, Mathura, India.-
E-mail: krishanveer.singh@gla.ac.in

Article History: Received: 11 January 2021; Accepted: 27 February 2021; Published online: 5 April 2021

Abstract: The "Online Dial-A-Ride Problem (ODARP)" is discussed in this article and time constraints and demand specifics include a deadline to reach the time limit. If a proposal is not filed by its delivery date, it will be cancelled. For data limit, only data on the origin is presented whenever the application is received public. And until the basis of the request is met, the target of the server is not statistics. The purpose of the problem is to schedule a server movement so that its deadlines can be met for a determined figure of queries (or the extreme amount of goods). We examine competitively several deterministic strategies with various constraints on the problem. And in various cases this paper shows many lower limits on the competitiveness of any deterministic algorithm for the problem.

Keywords: Online Dial-a-ride Problem, Competitive Analysis, Deadline, Information.

1. Introduction

DARP has been extensively investigated in the field of "organisational analysis, management science, and mixture optimization" because of its usefulness to the transportation and logistics sectors. In DARP, there are routers that travel through a number of metric areas to support bookings. For each ride, the position is specified in the topological space, source, start and destination. The issue is to build server routes through metric space in order to make all of the necessary rides and meet some criterion of optimality. Many new side limitations were applied to the issue to satisfy real-life needs. One important extension is the deadline problem of Dial-A-Ride. In other words, each application sets a deadline. If the application is not submitted by the due date, it will be withdrawn. The aim is to schedule the movement of servers so that their deadlines reach the full number of requests. Online requests are made over time during the natural setting when servers support other trips. And, once raised, the servers do not know at all times about potential requests. In other words, it's time to make and enforce decisions in online problem-setting. Taxi and minibus systems, postal services and elevators are examples of such issues in practice. In addition, an offline algorithm already has knowledge of all the applications for the entire list. The details on the request is usually known if a certain request is being sent. This ensures that the source and destination of the trip are clearly specified when presented during the release period. However, full ride requirements are not feasible in many practical circumstances. Sometimes only the ride source is provided when the application is released. Unless the server enters the source, it is not possible to obtain information about its destination. These models, e.g. the issues with the taxi or lift.[1-6]

Throughout this paper we are analyzing the ODARP for the first time with only a short deadline (this is called ODARPI). In ODARPI, a Server with some limited capacity travels in a metric space at a speed unit to satisfy a collection of travel requests that take items from resources to destinations. Increasing requirement has a time limit. When an application must be sent, the server must meet the point of origin of the request by its time limit. The purpose of the algorithm is to satisfy as many requirements by the deadlines as possible [7,9]. Until the application is published, the online algorithm for the problem does not know details. If a request is sent, all of the information on the request is made available to the online server (including the tool, destination, etc.). Secondly, the ODARP is analyzed with a time limit [8-11] and information limitation (reference is provided to ODARP2). Only information about the source is provided in this version of the problem when a request is submitted [12-16]. When the request source hits, the server does not have the information about the destination. Rides are called on as the server runs. Some products are there. The aim is to schedule the server's movements online in order to deliver the maximum number of products via request deadlines. For more than three decades ODARP and vehicle routing and scheduling issues were extensively studied (see (13) for a topic study). In most earlier studies on online routing issues, Based on the targets for the markup set [4, 5, 7], weighted completion times[17-24] and a maximum / average flow time [11, 13]. The document [6, 15, 16] presented the findings concerning the problem of online k-taxi scheduling, in which the request is based on a two-point graph or metric space (a source and a target). A related issue was subsequently explored in the K-truck online scheduling issue [25]. Both sides believe k servers (taxis or lorries) are free when a new request for service occurs and aim at the total distance of servers travel. (4) ODARP is researched in which rides are named during the server's journey. The authors have found two separate situations in which the server is able to do so and the server is able to do so. ODARP's first findings have been presented in a restricted knowledge model[14-18], which aims at reducing the period during which all the rides have been conducted on the server and returned to its root. All these works presumed that the requests should wait until they had been completed by the server [26 - 29]. In other words,

they did not respect the time limits. The offline view is the bulk of previous works on the DARP with time limits. The feedback is well understood in advance. See [8, 9, 31-33] for other linked plays. With the analysis of ODARPI, we are performing only a deadline test for REPLAN, a deterministic algorithm similar to BATCH algorithm in [17] which is competition with $4(1+34)$, when unit capacity is on the server, and -L -competitive when the server has infinite power, when I indicate a diameter of a general room' FCFS in uniform metric space, with 2Z and Creedy algorithms in K. First Come First-Serve algorithm (FCF Serving) Restricted metric space with 2ZR, where 2 are the server capacity and K shows the diameter for k Restricted metric space. For two versions of the problem, the remaining paper is arranged as follows, we also show a set of lower limits for the competitive ratio of any decisive algorithm [19-23, 27,29] We give some notes and assumptions in section 2. Section 3 presents the REPLAN algorithm and its competitive analysis on ODARPL Section 4 includes 2 online ODARP2 algorithms and two different types of metric spaces to discuss their performance [30-34].

2. Annotations and Symbols

Let $M = (X, d)$ be an undirected space induced by an undirected $G = (V, E)$ graph with $V = X'$ with n points, i.e. for any pair of M metric points, we have $d(x, y)$, which is the shortest G path between vertices of X and Y . In order to be in metric space between the two points, the diameter of a metric space is as high a time as possible as L . L . In this article we take into various metrical forms of space, coherent "metric space and K -confined metric space. The unit length of the standard metric space is the distance between two points. It can be regarded by the use of a full graph with unit range weights as a specific metric space. " $d_{\max} / d_{\min} = K$, where $d_{\max} = \max d(x, y)$, $d_{\min} = \min d(x, y)$, x, X, m, y, mt, V, mm ." We assume that d_{\min} equals 1 without losing the generality and d_{\max} / K is in the K -fixed metric field. So K is basically the diameter of the metric space that is confined to K so we can conclude that the server is moving in the metric space at speed 1. In uniform metric space, $K=1$ should be noted. The basic ODARP instance in the metric space M requires a sequence of applications $R: (r_1, r_2, \dots, r_m)$. Every request shall be fourfold " $r_i = (t_i, z_i, a_i, b_i) \in R, R \sim N = X$ THIS X " with: $T_i, \hat{c} \in R$ shall be the time that request r_i is made, shall be made public, and $Z_i \in N$ shall be the quantity of goods that shall be transmitted on the server, $A_i \in X$ and $B_i \in X$ are the source and destination, and the goods corresponding to the request r_i shall be transmitted on the server. The server capacity is finite and constant Z is indicated. This implies that the maximum limit of the loaded products on the server is two units. In ODARPI we assume z , for everyone I have 1 constant and Z an infinite value can be assumed. In ODARP2, we consider z omniscience, to be completely inaccurate. And the products of the applications are partial. That is to say, if a certain application has several products, it can be divided into many partitions in the sense of the server's current power. Such partitions can be loaded by the server. And the remainder can be regarded as the products of new demands. In this paper we assume that in order for non-discriminatory release times the sequence $R = (r_1, r_2, \dots, r_m)$ of the requests is given, i.e. $0 = \text{tot}1 \text{ fat } 2 \text{ complete alltm}$. An application is said to have been approved if the corresponding object is retrieved by the server on source and if the corresponding element is transported to the destination, the request will be completed. We are not permitting preemption: an authorized request is not permitted at any other location than its destination. This ensures it is not called off until a request is approved. Furthermore, the following assumptions are taken into account in this paper: a) Server speed is constant 1. This implies that the distance between the two points exactly is the time it takes to fly from one location to another; b) The deadline for all requests is standardized and is stated in T . In order to ensure that the problem is feasible, $T > d_{\min}$ in the metric space with K constraints and $T > 1$, in the metrically uniform space are assumed. By competitive analysis [1-3], which has evolved as a standard measure of success, we evaluate the efficiency of online algorithms. The output of an online algorithm is contrasted with the efficiency of the optimized offline algorithm that knows all future employment in competitive analysis. An algorithm A for ODARP is referred to as "highly competitive," where for any case, R is at least $1/\alpha$ more freight completed by A than by an optimal off line algorithm OPT . ODARP has the amount of freight supplied by A .

3. Algorithms

In this section, where "the Server has unit capacity and the infinite power of the Server in two separate cases, we give a RE PLAN algorithm for ODARPI. This section demonstrates the efficiency assurance for the algorithm. We take for granted that the deadline for the submission is $T=1$ in ODARPI for the purpose of analysis.

REPLAN Algorithm

Increasing step is divided into the basic plan with the REPLAN algorithm. We will take a certain time-span to fulfil a particular subset of requests at the start of the interval as the basic plan. The interval size is $1/2$. The server stops and replans at the start of each interval, providing a com with an ideal $1/2$ -long schedule starting from the current server location that the maximum number in the preceding interval of requests can be completed. Then the new fundamental plan continues to be used.

Theorem 1: “Provided that the server is unit-capacity ($Z = 1$), for any metric space with $L < 1/2$, REPLAN is $4\left(\frac{1+2L}{1-2L}\right)$ -competitive for ODARPT”.

Proof: The server has ability 1 i.e. at most 1 object can be transported simultaneously. If the server has accepted a request, it cannot accept another request until the approved request is completed. Let R_i be the set of all requirements arrived at interval I and let R_{ij} be R_i 's subset, finished at interval j . “Obviously, $R_{ij} \cap R_{ik} = \emptyset$ for all $j \in \{i, i+1, i+2, i+3\}$. Let $B_{opt}(R)$ denote the number of requests completed by OPT on a request sequence R . Suppose the last request arrives in m^{th} interval, then in the application series R , denote the number of requests completed by OPT”. Assume the last request enters m^{th} , then,

$$B_{opt}(R) = \sum_{i=1}^m (|R_{i,i}| + |R_{i,i+1}| + |R_{i,i+2}| + |R_{i,i+3}|) \quad (1)$$

Where $|R_{i,i}|, |R_{i,i+1}|, |R_{i,i+2}|, |R_{i,i+3}|$. At the start of interval I REPLAN carries out the main strategy, discovers a new plan and executes it. One way to RE is to choose the biggest choice of $R_{i-1,i-1}, R_{i-1,i}, R_{i-1,i+1}$ and $R_{i-1,i+2}$. At the start of an interval we distinguish between two cases depending on the OPT Server status.

The first is, at the beginning of any interval, that the OPT extreme is zero. Each one of these means $R_{i-1,i-1}, R_{i-1,i}, R_{i-1,i+1}$ and $R_{i-1,i+2}$. OPT may be served at one single period (accepted and fulfilled). This means there is a maximum $1/2$ round of rounds covering the entire $R_{i-1,i-1}$ field. The same is true for $R_{i-1,i}, R_{i-1,i+1}$ and $R_{i-1,i+2}$. Let $R_{max} = \{R | R = \max(|R_{i-1,i-1}|, |R_{i-1,i}|, |R_{i-1,i+1}|, |R_{i-1,i+2}|)\}$ be the set of requests which is the largest” of $R_{i-1,i-1}, R_{i-1,i}, R_{i-1,i+1}$ and $R_{i-1,i+2}$. The tour that covers R_{max} is marked by l_{max} . Now, at the highest possible starting point on the route l_{max} REPLAN's servers can take up to a certain time, and then it can take $1/2-L$ to satisfy further requests in R_{max} from an ideal location. The REPLAN server will be serving at least $(1-2L)$ requests during $1/2 - L$ times because it requires $1/2$ time to service R_{max} . Therefore at least $(1/2 - L)$ is the number of requests the REPLAN server will handle for interval i . We have denoted the total number of request sequence R by $B_{RE}(R)$ that REPLAN will serve,

$$B_{RE}(R) \geq (1 - 2L) \sum_{i=2}^{m+1} \max(|R_{i-1,i-2}|, |R_{i-1,i-1}|, |R_{i-1,i}|, |R_{i-1,i+1}|, |R_{i-1,i+2}|) \geq \left(\frac{1-2L}{4}\right) \sum_{i=1}^m (|R_{i,i}| + |R_{i,i+1}| + |R_{i-1,i+2}| |R_{i-1,i+2}|) = \frac{1-2L}{4} B_{OPT}(R) \quad (2)$$

Secondly, the server of the OPT actually holds an object for an application. It can be proved in the same way that

$$B_{RE}(R) \geq \frac{1}{4} \left(\frac{1-2L}{1+2L}\right) B_{OPT}(R) \quad (3)$$

So we can say REPLAN is $4\left(\frac{1+2L}{1-2L}\right)$ -competitive with unit-capacity for ODARPT in any $L < 1/2$ metric space.

Corollary 1: “Provided that the server has infinite capacity ($Z = \infty$), for any metric space with $L < 1/2$, REPLAN is $\frac{3}{1-2L}$ -competitive for ODARPT”.

The proofs of corollary 1 is similar to the proof of theorem 1.

4. Second Algorithms

In this part we show firstly that “there is no competitive deterministic algorithm for ODARPT in an unconstrained general metric space where time windows $T > 0$ are arbitrary. Then we study ODARPT2 with two unique types of metric space in a uniform metric space and in K -limited metric space. We give FCFS uniform metric space and greedy algorithm for K -controlled metric space. The effectiveness guarantees of the two problem algorithms are shown in this section.

Proposition: If the time limit $T > 0$ is arbitrary, a metric field in which a constant competitive ratio of ODARPT2 cannot be achieved by a deterministic online algorithm.

Proof: Take a metric space that includes a line. At 0, two requests are sent on the line with one product package. For a request, the server must load the product at position T and transmit it to position $2T$, where T is the request time period. The server also needs moving the products from position $-T$ to position $-2T$. Note, if the server does not arrive to the source in time T the requests will be shut down. If the online server does not leave for the requests immediately, the series will be stopped and no requests will be served. If not, we presume that the online server is in location $-T$ at the time T , without loss” of generality. “At the time when the off-line server enters T at T and delivers the items at $2T$ at $2T$ destination. Then the source position t and the destination

position $t+T(2+i)T$ ($i = 0,1,2, \text{ etc.}$) are displayed on request from time $t = 2T$ ". All requests can be handled by the off-line server but not by the on-line server.

4.1. FCFS Algorithms for ODARP2 in Uniform Metric Space

FCFS Algorithm

The FCFS algorithm works like this in a uniform metric space. The online server must always accept the original request first of all for a long time, receive data on the destination, collect the goods and then transfer them to the destination. If no definite requests can be processed in time after a particular request has been completed, the server stays at its current position and waits for new requests.

Theorem 2: FCFS algorithm is $2Z$ competitive with ODARP2 in the uniform metric space, where Z is the server's power.

Proof: Due to any input sequence R , these maximum sequences of m can always be divided into, " $R = (R_1, R_2, \dots, R_m)$ that in each sub-sequence $R_i = (r_{i,1}, r_{i,2}, \dots)$ ($1 \leq i \leq m$)", the online database operates simultaneously, i.e. it continuously answers R_i Applications one by one. We test an arbitrary R_i sequence for the first time and extend it to the whole R -series. We are going to show that in any subsequent series OPT cannot bring (or collect) up more than $2Z$ times the number of products according to the theorem.

Represent the time the first request " $r_{i,1}$ " was published. Let, " $u \in R_i$, and $r_{i,1} \in R$ ", be respectively the last re searches of FCFS and OPT. "Let t_{FCFS} be the time when FCFS reaches $r_{i,1}$'s travel destination and t_{OPT} refers to the time the OPT reaches $r_{i,1}$'s travel destination. We set t^* to ensure $(t_{OPT}-t^*)/2$ is a complete integral and $t_1 t^*$ to total $+ 2$. For t_{FCFS} and t_{OPT} ", there are two potential instances.

Case 1. "FCFS completes $r_{i,1}$ no later than time t_{opt} , i.e., $t_{FCFS} \leq t_{OPT}$. It can be proved that $t_{OPT}-t_{FCFS} < 2$. Provided that the subsequence R_i is maximal and that the server operates continuously in R_i , no further requests from R_i can be sent at $t > t_{fcfs}$. Just two units of time are required to collect and deliver the goods at their destination on one order at the source. In any time, therefore, " $(t, t + 2]$ for $t = t^*, t^* + 2, \dots, t_{OPT} - 2$ ", FCFS 'server may collect and/or deliver at a destination for at least a unit of goods while the OPT server is unable to collect and/or deliver more than $2Z$ units of goods at the source because of the finite capacity of Z . In the interval (t_1, t^*) , at least one request with one goods unit can be accepted on the FCFS server and no more than 2 requests (with $2Z$ units of goods) can be accepted on the OPT server, because $t^* - t_1$ We note that the number of goods accepted by the server (picked up)" is equal to the number of goods transported (delivered) since the prevention is not allowed. Therefore, we can tell the OPT to be no more than 27 times as much transiting goods as FCFS at any interval of " (t_1, t^*) and $(t, t+2)$ for $t = t^*, t+2, \dots, t_{opt} - 2$."

Case 2. FCFS still works after time " t_{opt} , i.e., $t_{fcfs} > t_{opt}$. Obviously, in each interval $(t, t + 2]$ for $t = t^*, t^* + 2, \dots, t_{OPT} - 2$, OPT can transport at most $2Z$ times as many goods as FCFS does. It also holds in the interval $(t_1, t^*]$ "

4.2 Greedy Algorithm for ODARP2 in K-Constrained Metric Space

Greedy Algorithm (GR)

The GR is described as "follows in the K -restricted metric space. Whenever the processor enters the metric space, the source of the request that is the least time path out of its current location will be identified amongst all the unknown requests, to be able to meet the goods within their timeframes, collect the goods and provide information on the destinations before they are over weighted. It then delivers to its destinations the products of the approved requests.

Theorem 3: The Greedy algorithm in the tightened metric space is $2ZK$ competitive for ODARP2, where Z is the server capability and K indicates the metric space diameter. There are a few shifts in the facts of theorem 3. The knowledge here is incomplete.

Corollary 2: GR is $2Z$ -competitive for ODARP2 in *the uniform metric space*".

5. Lower Bounds

In this section the competitive ratio for all deterministic online algorithms to fulfil the requests in the ODARPI and ODARP2 versions is less constrained. The results are achieved when the optimal algorithm is considered an opponent which defines the request sequence in such a way that the online algorithm performs poorly. However, the confirmation of the findings for page limits is omitted.

5.1 Lower Bounds for ODARPI

Theorem 4

Corollary 3: Provided the unit capacity ($Z=1$) of the server is available, for any $L < 1/2$, a metrical space of the diameter L can not be obtained with a competitive ratio less than $\frac{2-L}{2L} \left\lceil \frac{1}{L} \right\rceil$ in an online deterministic algorithm.

Where the server's capability is infinite ($Z = \infty$), for each $L < 1/2$, a metric space with a diameter L cannot be achieved in a competitive ratio below any deterministic inline algorithm. $\frac{2-L}{2L} \left\lceil \frac{1}{L} \right\rceil$ for ODARPI.

5.2 Lower Bounds for ODARP2

Theorem 5: There can be no competitive ratio of 2 for ODARP2, where 2 is server power, for the uniform metric space. Online algorithms can be deterministic.

Theorem 6: No deterministic online algorithm can achieve less competitive ratios of KZ in the K -restricted metric space than in ODARP2, where Z is the server's capacitance and K denotes the metric space's diameter.

6. Conclusions

In this paper we address the issues with time limits and knowledge in a wide range of realistic contexts online dial-a-ride. We are providing an online ODARPI algorithm in two kinds of metric space and two online ODARP2 algorithms. We evaluate REPLAN's efficiency for ODARPI, and FCFS for Upper Boundaries ODARP2 in "the uniform metric space, GR in the k -fixed metric space for ODARP2. For each problem in different situations, it would also be necessary to analyze other similar metric spaces (like trees, cycles) to test if better boundaries can be found. We also provide analysis for the lower limits of the competitive ratio of any deterministic" algorithms. The crew preparation in which more than one Server is used to handle the requests is an interesting extension of the problem.

References

7. Manasse, M.S., McGeoch, L.A., & Sleator, D.D. (1990). Competitive algorithms for server problems. *Journal of Algorithms*, 11(2), 208–230.
8. David, S.B., & Borodin, A. (1994). A new measure for the study of the on-line algorithm. *Algorithmic a*, 11(1), 73–91
9. Alon, N., Karp, R.M., & Peleg, D. (1995). A graph-theoretic game and its application to the k - server problem. *SIAM Journal on Computing*, 24(1), 78-100.
10. Kumar, A., Sharma, K., & Dixit, A. R. (2019). A review of the mechanical and thermal properties of graphene and its hybrid polymer nanocomposites for structural applications. *Journal of materials science*, 54(8), 5992-6026.
11. Feuerstein, E., & Stougie, L. (2001). On-line single server dial-a-ride problems. *Theoretical Computer Science*, 268(1), 91-105.
12. Sharma, K., & Shukla, M. (2014). Three-phase carbon fiber amine functionalized carbon nanotubes epoxy composite: processing, characterisation, and multiscale modeling. *Journal of Nanomaterials*, 2014.
13. Ascheuer, N., Krumke, S.O., & Rambau, J. (2000). Online dial-a-ride problems: Minimizing the completion time. In *Annual Symposium on Theoretical Aspects of Computer Science*, Springer, Berlin, Heidelberg, 639-650.
14. Yadav, A., Kumar, A., Singh, P.K., & Sharma, K. (2018). Glass transition temperature of functionalized graphene epoxy composites using molecular dynamics simulation. *Integrated Ferroelectrics*, 186(1), 106-114.
15. Xu, Y.F., & Wang, K.L. (1997). Scheduling for on-line taxi problem and competitive algorithms. *Journal of Xi'an Jiao Tong University*, 31(1), 56-61.
16. Sharma, K., Kaushalyayan, K.S., & Shukla, M. (2015). Pull-out simulations of interfacial properties of amine functionalized multi-walled carbon nanotube epoxy composites. *Computational Materials Science*, 99, 232-241.
17. Ausiello, G., Feuerstein, E., Leonardi, S., Stougie, L., & Talamo, M. (2001). Algorithms for the on-line travelling salesman 1. *Algorithmica*, 29(4), 560-581.
18. Psaraftis, H.N. (1983). An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows. *Transportation science*, 17(3), 351-357.

19. Diana, M., & Dessouky, M.M. (2004). A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows. *Transportation Research Part B: Methodological*, 38(6), 539-557.
20. Singh, P.K., Sharma, K., Kumar, A., & Shukla, M. (2017). Effects of functionalization on the mechanical properties of multiwalled carbon nanotubes: A molecular dynamics approach. *Journal of Composite Materials*, 51(5), 671-680.
21. Ma, W., Xu, Y., & Wang, K. (2001). On-line k-truck problem and its competitive algorithms. *Journal of Global Optimization*, 21(1), 15-25.
22. Singh, P.K., & Sharma, K. (2018). Molecular Dynamics Simulation of Glass Transition Behaviour of Polymer based Nanocomposites. *Journal of Scientific & Industrial Research*, 77(10), 592-595.
23. Hauptmeier, D., Krumke, S.O., & Rambau, J. (2000). The online dial-a-ride problem under reasonable load. *Lecture Notes in Computer Science*, 125-136.
24. Singh, P.K., & Sharma, K. (2018). Mechanical and viscoelastic properties of in-situ amine functionalized multiple layer graphene/epoxy nanocomposites. *Current Nanoscience*, 14(3), 252-262.
25. Krumke, S.O., De Paepe, W.E., Poensgen, D., & Stougie, L. (2003). News from the online traveling repairman. *Theoretical Computer Science*, 295(1-3), 279-294.
26. Shukla, M.K., & Sharma, K. (2019). Effect of carbon nanofillers on the mechanical and interfacial properties of epoxy based nanocomposites: A review. *Polymer Science, Series A*, 61(4), 439-460.
27. Krumke, S.O., Laura, L., Lipmann, M., Marchetti-Spaccamela, A., De Paepe, W., Poensgen, D., & Stougie, L. (2002). Non-abusiveness Helps: An O(1)-Competitive Algorithm for Minimizing the Maximum Flow Time in the Online Traveling Salesman Problem. In *APPROX*, 200-214.
28. Kumar, A., Sharma, K., & Dixit, A.R. (2020). Carbon nanotube-and graphene-reinforced multiphase polymeric composites: review on their properties and applications. *Journal of Materials Science*, 1-43.
29. Lipmann, M., Lu, X., De Paepe, W.E., Sitters, R.A., & Stougie, L. (2002). On-line dial-a-ride problems under a restricted information model. In *European Symposium on Algorithms, Springer, Berlin, Heidelberg*, 674-685.
30. Xu, Y.F., Wang, K.L., & Zhu, B. (1999). On the k-taxi problem. *Journal of Information*, 2(4), 429-434.
31. Xu, Y.F., Wang, K.L., & Ding, J.H. (1999). On-line k-taxi scheduling on a constrained graph and its competitive algorithm. *Journal of System Engineering (PR China)*, 4, 79-84.
32. Irani, S., Lu, X., & Regan, A. (2002). On-line algorithms for the dynamic traveling repair problem. In *Proceedings of the 13th Annual ACM/IEEE Symposium on Discrete Algorithms*, 517-524.
33. Bansal, P.K. (2020). Impact of Foreign Institutional Investors on the Sectorial Indices of National Stock Exchange. *Journal of Critical Reviews*, 7(7), 998-1002.
34. Agrawal, O.P., Bansal, P.K., & Kathpal, S. Effect of Financial Performance on Corporate Social Responsibility and Stock Price: A Study of BSE Listed Companies. *International Journal on Emerging Technologies* 11(1), 286-291.
35. Sharma, R., Awasthi, S., & Agarwal, K.K. (2020). Pilgrimage Destination and Expectations in India – A Study of Mathura and Vrindavan. *International Journal of Scientific & Technology Research*, 9(2), 4372-4377.
36. Jain, S. (2020). Impact of ICT on Enhancing Knowledge Sharing of Employees. *International Journal of Advanced Science and Technology*, 29(3), 262-280.
37. Jain, S. (2020). Employment expectations: a systematic review of the antecedents and their impact on the joining intention of prospective employees. *Journal of Critical Reviews*, 7(7), 512-516.
38. Ryu, H. (2019). Dynamics of SEIR Investment Model in Brokerage Business. *Asia-pacific Journal of Convergent Research Interchange*, 5(2), 69-77.
39. Harini, B., & Rao, T N. (2019). An Extensive Review on Recent Emerging Applications of Artificial Intelligence. *Asia-pacific Journal of Convergent Research Interchange*, 5(2), 79-88.
40. Kim, J. (2019). The Mediating Effect of Fear of Missing Out between Interpersonal Relationship Orientation and SNS Addiction Tendency in College Students. *Asia-pacific Journal of Convergent Research Interchange*, 5(3), 1-9.