



GLOBAL JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY
Volume 11 Issue 15 Version 1.0 September 2011
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Towards The Solution of Variants of Vehicle Routing Problem

By Pawan Jindal

Jaypee University of Engineering & Technology, Guna, M.P. India

Abstract - Some of the problems that are used extensively in real life are NP complete problems. There is no any algorithm which can give the optimal solution to NP complete problems in the polynomial time in the worst case. So researchers are applying their best efforts to design the approximation algorithms for these NP complete problems. Approximation algorithm gives the solution of a particular problem, which is close to the optimal solution of that problem. In this paper, a study on variants of vehicle routing problem is being done along with the difference in the approximation ratios of different approximation algorithms as being given by researchers and it is found that Researchers are continuously applying their best efforts to design new approximation algorithms which have better approximation ratio as compared to the previously existing algorithms.

Keywords : Approximation algorithms, Vehicle Routing problem with time windows, NP completeness.

GJCST Classification : C.2.2



Strictly as per the compliance and regulations of:



Towards The Solution of Variants of Vehicle Routing Problem

Pawan Jindal

Abstract - Some of the problems that are used extensively in real life are NP complete problems. There is no any algorithm which can give the optimal solution to NP complete problems in the polynomial time in the worst case. So researchers are applying their best efforts to design the approximation algorithms for these NP complete problems. Approximation algorithm gives the solution of a particular problem, which is close to the optimal solution of that problem. In this paper, a study on variants of vehicle routing problem is being done along with the difference in the approximation ratios of different approximation algorithms as being given by researchers and it is found that Researchers are continuously applying their best efforts to design new approximation algorithms which have better approximation ratio as compared to the previously existing algorithms.

Keywords : Approximation algorithms, Vehicle Routing problem with time windows, NP completeness.

1. INTRODUCTION

Transportation supports most of the social and economic activities. The annual cost of excess travel in U.S.A. has been estimated approximately 45 billion USD [68] and the turnover of transportation of goods in Europe is approximately 168 billion USD per year. In United Kingdom, France and Denmark transportation represents approximately 15%, 9% and 15% of national expenditures respectively [70]. It is well known fact that the vehicle routing problem is a combinatorial optimization and integer programming problem in which service to finite number of customers with a fleet of vehicles is being done. Vehicle routing problem was proposed by Dantzig and Ramser in 1959. Vehicle routing problem is an important optimization problem in the fields of distribution, transportation, and logistics. In Vehicle Routing problem, goods have to be delivered to the customers who have placed orders for such goods in such a way so that the total cost of the delivering of goods to the customers can be minimized. In VRP each and every customer has a given demand and no any vehicles can service more customers than its predefined capacity. Many algorithms have been designed by researchers for searching for good solutions to the problem, but no any polynomial time algorithms have been designed which can give the exact solution of Vehicle Routing problem with time windows in the polynomial time in the worst case. There are several variations of the vehicle routing problems. In

Vehicle Routing Problem with Pickup and Delivery, a large number of goods have to be moved from certain pickup locations to other delivery locations and the goal is to find optimal routes for vehicles to visit the pickup and drop-off locations so that the cost of delivering the goods to the different locations can be minimized. In Vehicle Routing Problem with LIFO principle, a large number of goods have to be moved from certain pickup locations to other delivery locations and the goal is to find optimal routes for vehicles to visit the pickup and drop-off locations so that the cost of delivering the goods to the different locations can be minimized with the restriction of the item being delivered must be that item most recently picked up. The benefit of this scheme over the previous scheme is the reduction of the loading and unloading times at delivery locations because there is no need to temporarily unload items. In Vehicle Routing Problem with Time Windows, there are n numbers of cities. Each and every city has a particular time windows $[R(v), D(v)]$. Where $R(v)$ represents the releasing time for a particular vertex and $D(v)$ represents the deadline for a particular vertex and the goal is to visit the maximum number of cities with in their time windows. Time windows are when they can be considered non bidding for penalty cost. Time windows are called hard when they cannot be violated, i.e. if any vehicle reaches to a particular city too early so it must wait unless and until the time windows opens and the vehicle is not allowed to arrive late. In Capacitated Vehicle Routing Problem, the vehicles have limited capacity of the goods that must be delivered. In Capacitated Vehicle Routing Problem with time windows $[R(v), D(v)]$, the vehicles have limited capacity of the goods that must be delivered. The most important application of VPRTW includes deliveries to supermarkets, industrial refuse collection, routing of school bus, security patrol services, urban newspaper distribution etc. The Vehicle Routing Problem with Time Windows has already been studied in the literature of Operations Research ([1, 10]). Different heuristics [9, 17, 18, 19] like Simulated Annealing, local search, Genetic algorithms, cutting plane and branch and bound methods [20, 14, 16] have been proposed to get the optimal solution of this problem. For general graphs, when there are a constant number of different time windows Chekuri and Kumar [8] gave a constant-factor approximation to solve vehicle routing problem with time windows. Capacitated Vehicle Routing Problem (CVRP) is the homogeneous VRP. In Multiple Depot VRP

Author : Department of Computer Science & Engineering, Jaypee University of Engineering & Technology, Guna, M.P. India.
E-mail : pawan_intelligent88@yahoo.co.in

(MDVRP) customers get their deliveries from several depots. In VRP with Time Windows [71] each and every customer has time window $(R[v], D[v])$ where $D[v]$ represents the deadline for a particular customer while $R[v]$ represents the releasing time for a particular customer and the goal is to visit the maximum number of customers in their time windows $(R[v], D[v])$. In Stochastic VRP (SVRP) any customer may have a random behavior. In Periodic VRP (PVRP) delivery to the customer is being done in some days. In Split Delivery VRP (SDVRP) several vehicles serve a customer. In the split delivery vehicle routing problem (SDVRP) there is no any restriction of visiting each and every customer exactly visited once.

Additionally, the demand of each and every customer may be greater than the capacity of the vehicles. It is well known fact that the SDVRP is NP-hard problem, even under restricted conditions on the costs, when each and every vehicle have a capacity greater than two. But it can be solved in the time complexity of polynomial time when the vehicles have a maximum capacity of two. The cost saving that which can be obtained by allowing split deliveries can be up to 50% of the cost of the optimal solution of the VRP.

The variant of the VRP[71] in which the demand of a customer may be greater than the vehicle capacity, but vehicle has to serve every customer minimum number of the possible time. The cost saving which can be obtained by allowing more than the minimum number of required visits to each and every customer to be served by vehicle can be again up to 50%. Simple heuristics that serve the customers with demands greater than the vehicle capacity by full load out-and-back trips until the demands become less than the vehicle capacity may be quite far from the optimal solution.

Three heuristic methods [71] have been already proposed for the solution of the SDVRP: The local search, a simple and effective tabu search algorithm and a sophisticated heuristic which uses the information collected during the tabu search, builds promising routes and solves MILP models to decide which routes to use and how to serve the customers through those routes to obtain the solution which is close to the optimal solution. The heuristics will be compared on a set of benchmark instances.

In VRP with Backhauls (VRPB) vehicle must pick something up from the customer after all deliveries are done to the customers. VRP with Backhauls (VRPB) is also known as the linehaul-backhaul problem which is an extension of the Capacitated VRP (CVRP) where the customer set is partitioned into two subsets. The first subset contains the linehaul customers; each requires a given quantity of product which has to be delivered. The second subset contains the backhaul customers, where a given quantity of inbound product must be picked up. This customer partition is extremely frequent in practical

situations. Grocery industry is a common example, where supermarkets and shops are the linehaul customers and grocery suppliers are the backhaul customers. It has been widely recognized that in this mixed distribution—collection context a significant saving in transportation costs can be achieved by visiting backhaul customers in distribution routes. More precisely, the VRPB [71] can be stated as the problem of determining a set of vehicle routes visiting all customers, and (a) each vehicle performs exactly one route; (b) each route starts as well as finishes at the depot; (c) for each route the total load associated with linehaul and backhaul customers should never exceed, separately, the vehicle capacity; (d) on each route the backhaul customers, are visited after all linehaul customers; and (e) the total distance traveled by the vehicles to serve the customers is minimized. The constraint (d) is practically motivated by the fact that vehicles are rear loaded which proves that the onboard load rearrangement required by a mixed service is difficult to carry out at customer locations. The most important reason is that, in many applications, line haul customers have a higher service priority as compared to backhaul customers. In VRP with Pick-Ups and Deliveries (VRPPD) the vehicle picks something up and delivers it to the customer.

II. VEHICLE ROUTING PROBLEM WITH TIME WINDOWS

If $V[72]$ represents set of vehicles and all vehicles are considered to be identical. C represents set of customers. $G=(N, A)$ represents directed graph where N represents the set of vertices of graph while E represents the set of edges of graph. This particular directed graph consists of $|C|+2$ number of vertices, where customers are denoted $1,2,3,\dots,n$ and the depot is represented by the vertex "0"(the starting depot) and the vertex " $n+1$ "(the returning depot). N is the set of vertices. There is no edge ending at the vertex "0" or originating from the vertex " $n+1$ ". c_{ij} (where $i \neq j$) represents the cost of traveling from the vertex i to the vertex j . t_{ij} (where $i \neq j$) represents the service time at the customer i . Each vehicle has a capacity q and each customer i has a demand d_i . Each and every customer has a time window $[a_i, b_i]$ and a vehicle must arrive at the customer before b_i . If any vehicle arrives to the customer before the time windows opens, that vehicle has to be wait until a_i to service the customer. The time windows for both depots are assumed to be identical to $[a_0, b_0]$ which represents the scheduling horizon. The vehicle cannot leave the depot before a_0 and must return at the latest at time b_{n+1} . It is assumed that $q, a_i, b_i, d_i, c_{ij}, t_{ij}$ are positive integers. It is also assumed that the triangle inequality [72] is satisfied for both c_{ij} and t_{ij} . There are two sets of decision variables x and s . For each edge (i,j) where $i \neq j, i \neq n+1, j \neq 0$ and each vehicle k we define x_{ijk} as

$$X_{ijk} = \begin{cases} 1, & \text{if vehicle } k \text{ drives directly from vertex } i \text{ to the vertex} \\ 0, & \text{otherwise.} \end{cases}$$

The decision variable s_{ik} is defined for each and vertex i and each vehicle k and denoted the time when the vehicle k starts to service the customer i . When the vehicle k does not service to the customer i , s_{ik} has no meaning and consequently its value is considered irrelevant. As we have assumed $a_0=0$ and therefore $s_{0k}=0$ for all k . The goal in the case of VRPTW is to design a set of routes that minimizes the total cost, such that 1, if vehicle k drives directly from vertex i to the vertex

- (a) Each customer is being served by vehicle exactly once.
- (b) Every route starts at the vertex 0 and ends at vertex $n+1$.
- (c) The time windows of the customers and capacity constraints of the vehicles are being observed carefully.

The above informal definition of VRPTW can be stated mathematically as a multicommodity network flow problem with time windows and the capacity constraints:

$$\min \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij} X_{ijk} \text{ such that} \quad (1)$$

$$\sum_{k \in V} \sum_{j \in N} X_{ijk} = 1 \quad \forall i \in C \quad (2)$$

$$\sum_{i \in C} d_i \sum_{j \in N} X_{ijk} \leq q \quad \forall k \in V \quad (3)$$

$$\sum_{j \in N} X_{0jk} = 1 \quad \forall k \in V \quad (4)$$

$$\sum_{i \in N} X_{ihk} - \sum_{j \in N} X_{hjk} = 0 \quad \forall h \in C, \forall k \in V, \quad (5)$$

$$\sum_{j \in N} X_{i,n+1,k} = 1 \quad \forall k \in V \quad (6)$$

$$X_{ijk} (s_{ik} + t_{ij} - s_{jk}) \leq 0 \quad \forall i, j \in N, \forall k \in V \quad (7)$$

$$a_i \leq s_{ik} \leq b_i \quad \forall i \in N, \forall k \in V \quad (8)$$

$$X_{ijk} \in \{0, 1\} \quad \forall i, j \in N, \forall k \in V \quad (9)$$

The main goal of the objective function (1) is to minimize the total travel cost. The constraint (2) ensures that each customer is visited exactly once while constraint (3) ensures that a vehicle can only be loaded up to its capacity. Equations 4 indicated that each and every vehicle must leave the depot 0. Equation 5

indicates that when a vehicle arrives at a customer it must leave for another destination. Equation 6 indicates that all vehicles must arrive at the depot $n+1$. Inequality (7) indicates the relationship between the departure time of vehicle from the customer and its immediate successor. Constraint (8) indicates the observation of time windows. Integrality constraints are shown by (9). The model for the representation of VRPTW can also incorporate a constraint giving an upper bound on the number of vehicles, as is the case in Desrosiers, Dumas, Solomon and Soumis [59].

III. VEHICLE ROUTING PROBLEM WITH TIME WINDOWS IS NP COMPLETE PROBLEM

Sorting algorithms like selection sort, bubble sort, insertion sort are known as quadratic sorting because these algorithms have time complexity of $O(n^2)$ in the worst case where n is the size of input. Sorting algorithms like counting sort, radix sort and bucket sort have linear time complexity in the worst case. So these algorithms are known as sorting in linear time. It is well known fact that all problems cannot be solved in polynomial time in the worst case. For example, Turing's famous "Halting Problem," which cannot be solved by any computer, no matter how much time is provided. Those problems that can be solved, but in time $O(n^k)$ for any constant k are known as tractable problems or easy problems. For example sorting algorithms like selection sort, bubble sort, insertion sort, counting sort, radix sort and bucket sort can give sorted output in the time complexity of polynomial time so sorting problems are tractable problems or easy problems. Those problems that can not be solved, in polynomial time $O(n^k)$ for any constant k but they require super-polynomial time for their executions are known as intractable problems or hard problems. No polynomial-time algorithm has yet been discovered for an NP-complete problem which can solve the problem in the polynomial time in the worst case nor anyone has been able to prove that no polynomial-time algorithm can exist for any one of NP-complete problems. So $P \neq NP$ question has been one of the deepest, most perplexing open research problems in theoretical computer science since it was first proposed in 1971.

The class P consists of those problems which can be solved in polynomial time in the worst case. More specifically, they are problems that can be solved in time $O(n^k)$ for some constant k , where n is the size of the input to the problem. Problems like sorting problem, searching problems are in class P . The class NP consists of those problems that are "verifiable" in polynomial time. If a "certificate" of a solution is being given, then we could verify that the certificate is correct in time polynomial in the size of the input to the problem. Hamiltonian-cycle problem is in class NP because In Hamiltonian-cycle problem, directed graph $G = (V, E)$ is being given, and then certificate of this problem would

be a sequence $v_1, v_2, v_3, \dots, v_n$ of $|V|$ vertices. It is easy to check in polynomial time whether these set of vertices would be lead to the Hamiltonian cycle or not. Also, 3-CNF satisfiability problem is in class NP. It is well known fact that any problem in class P will also in class NP, because if a problem is in P then we can solve it in polynomial time without even being given a certificate. Any problem will be lie in the class NPC-and we refer to it as being NP-complete-if the problem is in NP and is as "hard" as any problem in NP. The first NP complete problem is the circuit-satisfiability problem, in which we are given a Boolean combinational circuit which is being consists of AND, OR, and NOT gates and the question is to know whether there is any set of Boolean inputs to this circuit that causes its output to be 1. It is well known fact that the concept of NP-complete was firstly introduced by Stephen Cook in 1971 in a paper entitled "The complexity of theorem-proving procedures" on pages 151-158 of the Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, Although the term NP-complete did not appear anywhere in his paper. At that conference, there was a debate among the computer researchers about whether NP-complete problems could be solved in polynomial time on a deterministic Turing machine or not. At that time, John Hopcroft brought everyone at the conference to a consensus that the question of whether NP-complete problems are solvable in polynomial time or not must be put off to be solved at some later time, since nobody had any formal proofs for their claims. No any scientist has yet been able to prove conclusively whether NP-complete problems are solvable in polynomial time or not. Also The Clay Mathematics Institute is offering a US\$1 million reward to any researcher who has a formal proof that $P=NP$ or that $P \neq NP$. Researchers are continuously doing hard work in this field to give the formal prove of either $P=NP$ or $P \neq NP$ but did not achieve success in this field till date. Also In the celebrated Cook-Levin theorem, Cook proved that the Boolean satisfiability problem is NP-complete problem. In 1972, Richard Karp proved that several other problems are also NP-complete; So it shows that there is a class of NP-complete problems. Satisfiability, 0-1 Integer Programming, Clique, Set Cover, Vertex Cover, Set Covering, Feedback Node Set, Feedback Arc Set, Directed Hamilton Circuit Undirected Hamilton Circuit, Satisfiability With At Most 3 Literals Per Clause, Chromatic Number, Cover, Exact, Hitting, Steiner Tree, 3-Dimensional Matching Knapsack (Karp's definition of Knapsack is closer to Subset sum), Job Sequencing, Partition Max Cut. After then, thousands of other problems have been shown to be NP-complete problems by reductions from other problems previously shown to be NP-complete. There is no any algorithm which can solve Vehicle Routing Problem with time windows in the polynomial time in the worst case. It is well known fact that Vehicle Routing Problem with time

windows is NP complete problem. So researchers are continuously applying their best efforts to give approximation algorithm for Vehicle Routing Problem with time windows. As it well known fact that approximation algorithms give the approximation result which is close to the optimal value to a particular problem. Most of the problems of practical significance are NP-complete but we cannot avoid them. There are three approaches to getting around NP-completeness of the problems. First, if the size of inputs is small, an algorithm which has exponential running time may be a satisfactory algorithm to solve a problem. Second, we may isolate those special cases that are solvable in polynomial time. Third, we can find out the solution which is close to the optimal solution of the problem and that solution can be easily found out with the help of the polynomial time approximation algorithm. Suppose there is an optimization problem in which each potential solution has a positive cost. It is well known fact that the optimization problem can be divided in the two parts. The maximization problem and the minimization problem. In maximization problem, we want to maximize the value of output to a problem and in minimization problem we want to minimize the value of output. If the size of the input of a problem is n . Let C^* be the cost as being obtained by optimal solution of a problem and C is the cost as being obtained by approximation algorithm [49]. Then the approximation ratio $\rho(n)$ is defined as

$$\text{Maximum } (C/C^*, C^*/C) \leq \rho(n).$$

The definitions of approximation ratio and of $\rho(n)$ -approximation algorithm apply for both minimization and maximization problems. It is well known fact that for a maximization problem, $0 < C \leq C^*$ and the ratio C^*/C gives the ratio by which the cost of optimization algorithm is larger than that of the cost of the approximate solution. Also, for a minimization problem, $0 < C^* \leq C$, and the ratio C/C^* gives the ratio by which the cost of the approximate solution is greater than the cost of an optimal solution. Since all solutions are assumed to have positive cost, these ratios are always well defined. Also the approximation ratio of an approximation algorithm [49] is never less than 1. For some problems, there are polynomial-time approximation algorithms which have small constant approximation ratios, but for other problems, the best known polynomial-time approximation algorithms have approximation ratios that grow as functions of the size of input of the problem.

An approximation scheme for an optimization problem [49] is defined as an approximation algorithm which takes as input an instance of the problem, along with a value $\epsilon > 0$ such that for any fixed, the scheme is a $(1 + \epsilon)$ -approximation algorithm. An approximation Scheme is said to be a polynomial-time approximation scheme if and only if for any fixed $\epsilon > 0$, the scheme runs in polynomial time in the size n of its input instance.

IV. COMPARISON OF APPROXIMATION ALGORITHMS FOR VEHICLE ROUTING PROBLEM

Arkin, Mitchell and Narasimhan [26] gave first non-trivial $(2 + \varepsilon)$ approximation algorithm for orienteering points in the Euclidean plane. A. Blum, S. Chawla, D. Karger, T. Lane, A. Meyerson, and M. Minkoff [6] gave the first approximation algorithm with a ratio of 4 for points in arbitrary metric spaces. After then N. Bansal, A. Blum., S. Chawla, and A. Meyerson [24] designed a new approximation algorithm for orienteering problem which improved this ratio to 3. A related problem to the orienteering problem is the minimum excess problem as being defined in [6]. In [6] the pseudo code for the orienteering problem depends upon the pseudo code of the min-excess problem. Also the min-excess problem can be approximated using algorithms for the k-stroll problem. In the k-stroll problem, the goal is to find a minimum length walk from source vertex s to target vertex t that visits at least k vertices. It is well known fact that k-stroll problem and the orienteering problem are equivalent to each other in terms of exact solvability because in both of these problems, the mission is to find the minimum length path from the source vertex s to the destination vertex t which covers maximum number of distinct vertices. The results in [6, 24] are based on existing approximation algorithms for k-stroll in undirected graphs. N. Bansal, A. Blum, S. Chawla and A. Meyerson.[24] gave approximation algorithm for Deadline-TSP which has the time complexity of $O(\log n)$ and they gave approximation algorithm for Vehicle Routing problem with time windows which has time complexity of $O(\log^2 n)$. Further they gave a bicriteria approximation algorithm for Deadline-TSP as well as for Vehicle Routing problem with time windows. If $\varepsilon > 0$, their bicriteria approximation algorithm produces a $\log(1/\varepsilon)$ approximation, while deadlines exceeds by a factor of $(1 + \varepsilon)$. C. Chekuri, N. Korula, and M. Pal [42] designed $(2 + \varepsilon)$ approximation for orienteering in undirected graphs, which improves upon the 3-approximation of [24]. C. Chekuri, N. Korula, and M. Pal [42] designed an improved $O(\log^2 \text{OPT})$ approximation for orienteering in directed graphs, where $\text{OPT} \leq n$ is the number of vertices visited by an optimal solution which improves over the previously result. Further it was being proved that for the time-window problem, an $O(\log \text{OPT})$ approximation can be easily achieved even for directed graphs if the algorithm is allowed quasi-polynomial time. As it has been already discussed that the best known polynomial time approximation ratios for Vehicle Routing problem with time windows are $O(\log^2 \text{OPT})$ for undirected graphs and $O(\log^4 \text{OPT})$ in directed graphs. If $D(v)$ represents the deadline for a particular vertex v and $R(v)$ represents the releasing time for a particular vertex v . Let $L(v) = D(v) - R(v)$ denotes the length of the

time-window for the vertex v and let $L_{\max} = \max_v L(v)$ and $L_{\min} = \min_v L(v)$. Let a be the known approximation ratio for orienteering problem. As $a = O(\log^2 \text{OPT})$ for directed graphs. C. Chekuri and N. Korula. Designed an $O(a \log L_{\max})$ approximation when $R(v)$ and $D(v)$ both are integer valued for each v and they designed an $O(a \max\{\log \text{OPT}, \log L_{\max}/L_{\min}\})$ approximation. They also designed an $O(\log L_{\max}/L_{\min})$ approximation when there is no starting vertex and terminating vertex is being defined. Early surveys of solution techniques for the VRPTW[67] can be found in Golden and Assad [57], Desrochers et al. [58], and Chiang & Russell[66]. Desrosiers et al. [59] and Cordeau et al.[60] gave exact solution techniques for VRPTW. The complete explanation of these exact techniques can be found in Larsen [61] and Cook and Rich [62]. Researchers designed different approximation algorithms for VRPTW based on different designing techniques like Dynamic programming, Simulated Annealing etc. Fleischmann [63] and Taillard et al.[64] have used heuristic for VRP without time windows. In Taillard et al. [64], have designed solutions to the classical vehicle routing problem by using a TS heuristic. The routes which are obtained combine to produce workdays for the vehicles by solving a bin packing problem, an idea which is previously introduced in Fleischmann [63]. Compbell and Savelsbergh [65] has reported about insertion heuristics which can efficiently handle different types of constraints including time windows and multiple uses of vehicles. Compbell and Savelsbergh [65] introduced the home delivery problem which is the variant of Vehicle Routing problem and it is more closely related to real-world applications. Current VRPTW heuristics can be categorized as follows: (i) construction heuristics, (ii) improvement heuristics and (iii) meta-heuristics. Construction heuristics are sequential or parallel algorithms which aims at designing initial solutions to routing problems that can be easily improved upon by meta-heuristics or improvement heuristics. Sequential algorithms are being used to build a route for each vehicle, one after another with the help of decision functions for the selection of the customer which has to be inserted in the route and the insertion position within the route. Parallel algorithms build the routes for all vehicles in parallel by using a pre-computed estimate of the number of routes.

V. CONCLUSIONS

In this paper, a study on variants of vehicle routing problem is being done along with the difference in the approximation ratios of approximation algorithm as being given by researchers and it is found that Researchers are continuously applying their best efforts to design new approximation algorithms which have better approximation ratio as compared to the previously existing approximation algorithms. Researchers are proposing new heuristics for variants of Vehicle Routing problems.

REFERENCES REFERENCES REFERENCIAS

1. E.M. Reingold, J. Nievergelt and N. Deo, "Combinatorial Algorithms: Theory and Practice", Prentice-Hall, 1977.
2. Allahverdi, J. Gupta and T. Aldowaisan. A review of scheduling research involving setup considerations. *Omega, Int. Journal of Management Science*, 27:219–239, 1999.
3. E. M. Arkin, J. S. B. Mitchell, and G. Narasimhan. Resource-constrained geometric network optimization. In *Symposium on Computational Geometry*, pages 307–316, 1998.
4. B. Awerbuch, Y. Azar, A. Blum, and S. Vempala. Improved approximation guarantees for minimum-weight k-trees and prizecollecting salesmen. *Siam J. Computing*, 28(1):254–262, 1999.
5. R. Bar-Yehuda, G. Even, and S. Shih. On approximating a geometric prize-collecting traveling salesman. In *Proceedings of the European Symposium on Algorithms*, 2003.
6. Blum, S. Chawla, D. Karger, T. Lane, A. Meyerson, and M. Minkoff. Approximation algorithms for orienteering and discounted-reward tsp. In *Proceedings of the 44th Foundations of Computer Science*, 2003.
7. J. Bruno and P. Downey. Complexity of task sequencing with deadlines, set-up times and changeover costs. *SIAM Journal on Computing*, 7(4):393–404, 1978.
8. G.N. Frederickson and B. Wittman. Approximation algorithms for the traveling repairman and speeding deliveryman problems with unit-time windows. *Proceedings of APPROX-RANDOM*, LNCS 4627:119–133, 2007.
9. M. Desrochers, J. Desrosiers, and M. Solomon. A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, 40:342–354, 1992.
10. M. Desrochers, J. Lenstra, M. Savelsbergh, and F. Soumis. Vehicle routing with time windows: optimization and approximation. In B.L. Golden, A.A. Assad (eds.), *Vehicle Routing: Methods and Studies*, North-Holland, Amsterdam, pages 65–84, 1988.
11. B. Golden, L. Levy, and R. Vohra. The orienteering problem. *Naval Research Logistics*, 34:307–318, 1987.
12. M. Gravel, W. Price, and C. Gagn. Scheduling jobs in a alcan aluminium factory using a genetic algorithm. *International Journal of Production Research*, 38(13):3031–3041, 2000.
13. C. Jordan. A two-phase genetic algorithm to solve variants of the batch sequencing problem. *International Journal of Production Research (UK)*, 36(3):745–760, 1998.
14. M. Kantor and M. Rosenwein. The orienteering problem with time windows. *Journal of the Operational Research Society*, 43:629–635, 1992.
15. Y. Karuno and H. Nagamochi. A 2-approximation algorithm for the multi-vehicle scheduling problem on a path with release and handling times. In *Proceedings of the European Symposium on Algorithms*, pages 218–229, 2001.
16. Kolen, A. R. Kan, and H. Trienekens. Vehicle routing with time windows. *Operations Research*, 35:266–273, 1987.
17. S. Thangiah. Vehicle Routing with Time Windows using Genetic Algorithms, *Application Handbook of Genetic Algorithms: New Frontiers*, Volume II. Lance Chambers (Ed.). CRC Press, 1995.
18. K. Tan, L. Lee, and K. Zhu. Heuristic methods for vehicle routing problem with time windows. In *Proceedings of the 6th AI and Math*, 2000.
19. M. Savelsbergh. Local search for routing problems with time windows. *Annals of Operations Research*, 4:285–305, 1985.
20. S. R. Thangiah, I. H. Osman, R. Vinayagamoorthy and T. Sun. Algorithms for the vehicle routing problems with time deadlines. *American Journal of Mathematical and Management Sciences*, 13(3&4):323–355, 1993.
21. J. Tsitsiklis. Special cases of traveling salesman and repairman problems with time windows. *Networks*, 22:263–282, 1992.
22. J. Wisner and S. Siferd. A survey of u.s. manufacturing practices in make-to-order machine shops. *Production and Inventory Management Journal*, 1:1–7, 1995.
23. W. Yang and C. Liao. Survey of scheduling research involving setup times. *International Journal of Systems Science*, 30(2):143–155, 1999.
24. N. Bansal, A. Blum., S. Chawla, and A. Meyerson. Approximation algorithms for deadline-TSP and vehicle routing with time-windows. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 166–174. ACM New York, NY, USA, 2004.
25. R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Upper Saddle River, New Jersey, 1993.
26. E.M. Arkin, J.S.B. Mitchell, and G. Narasimhan. Resource-constrained geometric network optimization. In *Symposium on Computational Geometry*, pages 307–316, 1998.
27. S. Arora and G. Karakostas. A $2 + \epsilon$ approximation algorithm for the k- MST problem. *Mathematical Programming A*, 107(3):491–504, 2006. Preliminary version in *Proc. of ACM-SIAM SODA*, 754–759, 2000.
28. B. Awerbuch, Y. Azar, A. Blum, and S. Vempala. Improved approximation guarantees for minimumweight k-trees and prizecollecting

- salesmen. SIAM J. Computing, 28(1):254–262, 1998. Preliminary Version in Proc. of ACM STOC, 277–283, 1995.
29. E. Balas. The prize collecting traveling salesman problem. *Networks*, 19(6):621–636, 1989.
 30. A.M. Frieze, G. Galbiati, and F. Maffioli. On the worst-case performance of some algorithms for the asymmetric traveling salesman problem. *Networks*, 12(1):23–39, 1982.
 31. R. Bar-Yehuda, G. Even, and S. Shahar. On approximating a geometric prize-collecting traveling salesman problem with time windows. *Journal of Algorithms*, 55(1):76–92, 2005. Preliminary version in Proc. Of ESA, 55–66, 2003.
 32. N. Garg. A 3-approximation for the minimum tree spanning k vertices. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, pages 302–309. IEEE Computer Society, 1996.
 33. Blum, R. Ravi, and S. Vempala. A Constant-Factor Approximation Algorithm for the k -MST Problem. *Journal of Computer and System Sciences*, 58(1):101–108, 1999. Preliminary version in Proc. Of ACMSTOC, 1996.
 34. K. Chaudhuri, B. Godfrey, S. Rao, and K. Talwar. Paths, trees, and minimum latency tours. In *44th Annual Symposium on Foundations of Computer Science*, pages 36–45. IEEE Computer Society, 2003.
 35. C. Chekuri and A. Kumar. Maximum coverage problem with group budget constraints and applications. *Proceedings of APPROXRANDOM*, pages 72–83, 2004.
 36. C. Chekuri and M. P'al. A recursive greedy algorithm for walks in directed graphs. In *Proceedings of the 46th Annual Symposium on Foundations of Computer Science*, pages 245–253. IEEE Computer Society, 2005.
 37. C. Chekuri and M. P'al. An $O(\log n)$ Approximation for the Asymmetric Traveling Salesman Path Problem *Theory of Computing*, 3:197–209, 2007. Preliminary version in Proc. of APPROX, 95–103, 2005.
 38. N. Garg. Saving an epsilon: a 2-approximation for the k -MST problem in graphs. In *Proceedings of the 37th Annual ACM Symposium on Theory of computing*, pages 396–402. ACM, 2005.
 39. M. X. Goemans and D. P. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24:296–317, 1995.
 40. B.L. Golden, L. Levy, and R. Vohra. The orienteering problem. *Naval Research Logistics*, 34(3):307–318, 1987.
 41. G. Gutin and A.P. Punnen, editors. *The traveling salesman problem and its variations*. Springer, Berlin, 2002.
 42. C. Chekuri, N. Korula, and M. P'al. Improved Algorithms for Orienteering and Related Problems. In *Proc. of ACM-SIAM SODA*, January 2008.
 43. E.L. Lawler, A.H.G. Rinnooy Kan, J.K. Lenstra, and D.B. Shmoys, editors. *The Traveling salesman problem: a guided tour of combinatorial optimization*. John Wiley & Sons Inc, 1985.
 44. Nagarajan and R. Ravi. Poly-logarithmic approximation algorithms for directed vehicle routing problems. *Proceedings of APPROXRANDOM*, LNCS 4627:257–270, 2007.
 45. P. Toth and D. Vigo, editors. *The vehicle routing problem*. SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial Mathematics, Philadelphia PA, 2001.
 46. J.N. Tsitsiklis. Special cases of traveling salesman and repairman problems with time windows. *Networks*, 22(3):263–282, 1992.
 47. K. Chen and S. Har-Peled. The orienteering problem in the plane revisited. *SIAM J. on Computing*, 38(1):385–397, 2008. Preliminary version in Proc. of ACM SoCG, 247–254, 2006.
 48. E. Horowitz and S. Sahni, "Fundamental of Computer Algorithms", Computer Science Press, 1982.
 49. T. Cormen, C. Leiserson and R. Rivest, "Introduction to Algorithms", MIT Press, 1991.
 50. A.V. Aho, J.E. Hopcroft and J.D. Ullman, "The Design and Analysis of Computer Algorithms", Addison-Wesley, 1974.
 51. A.V. Aho, J.E. Hopcroft and J.D. Ullman, "Data Structures and Algorithms", Addison-Wesley, 1984.
 52. S. Baase, "Computer Algorithms: Introduction to Design and Analysis", Addison-Wesley 1988.
 53. L. Banachowski, A. Kreczmar and W. Rytter, "Analysis of Algorithms and Data Structures", Addison-Wesley, 1991.
 54. R. Baeza-Yates, "Teaching Algorithms", IV Iberoamerican Congress on Computer Science Education, Canela, Brazil, July 1995.
 55. G. Tel, "Introduction to Distributed Algorithms" Cambridge Univ. Press 1995.
 56. G- Brassard, and P. Bratley, "Algorithmics: Theory and Practice", Prentice-Hall, 1988.
 57. Bruce L. Golden, Arjang A. Assad, Edward A. Wasil: Introduction. *Computers & OR* 13(2-3): 107-108 (1986).
 58. *Vehicle Routing with Time Windows: Optimization and Approximation* – Desrochers, Lenstra, et al. – 1988.
 59. *Time constrained routing and scheduling* – Desrosiers, Dumas, et al. – 1995.
 60. Cordeau, G. Laporte and A. Mercier. "A unified tabu search heuristic for vehicle routing problems with time windows." *Journal of the Operational Research Society*, 52:928–936 (2001).
 61. Larsen J. 1999. "Parallelization of the Vehicle Routing Problem with Time Windows" Ph.D. thesis, Institute of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark.

62. Cook W. & Rich J.L. 1999. "A Parallel Cutting-Plane Algorithm for the Vehicle Routing Problems with Time Windows." Working Paper, Department of Computational and Applied Mathematics, Rice University, Houston, U.S.A.
63. Fleischmann B. 1990. "The vehicle routing problem with multiple use of vehicles." Working Paper. Fachbereich Wirtschaftswissenschaften, Universität Hamburg, Germany.
64. Taillard E.D., Laporte G. & Gendreau M. 1996. "Vehicle routing with multiple use of vehicles" *Journal of the Operational Research Society* 47: 1065–1070.
65. Campbell A.M. & Savelsbergh, M. 2004. "Efficient insertion heuristics for vehicle routing and scheduling problems" *Transportation Science* 38(3), 369–378.
66. Chiang W. & Russell R.A. 1996. "Simulated annealing metaheuristics for the vehicle routing problem with time windows." *Annals of Operations Research* 63: 3-27.
67. Liong Choong Yeun, Wan Rosmanira Ismail, Khairuddin Omar2 & Mourad Zirour 2008. "Vehicle Routing Problem: Models and Solutions *Journal of Quality Measurement and Analysis*" Volume 4, Issue 1, pp. 205-218.
68. King, G.F. and C.F. Mast (1997), "Excess Travel: Causes, Extent and Consequences", *Transportation Research Record* 1111, 126–134.
69. Crainic, T. G. and G. Laporte (1997), "Planning Models for Freight Transportation", *European Journal of Operational Research* 97, 409–438.
70. Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms.
71. http://osiris.tuwien.ac.at/~wgarn/VehicleRouting/vehicl_routing.html
72. <http://alvarestech.com/temp/vrptw/Vehicle%20Routing%20Problem%20with%20Time%20Windows.pdf>