CHAPTER 3

LITERATURE REVIEW

3.1 INTRODUCTION

VRPTW and MDVRPTW belong to the class of Non-deterministic Polynomial (NP) hard combinatorial optimisation problems. Because of the high complexity level of VRPTW and MDVRPTW and its widespread applicability in real life situations, the development of heuristics which are capable of producing high quality solutions is of prime importance. This chapter reviews the existing algorithms to solve VRPTW and MDVRPTW.

3.2 SINGLE DEPOT VEHICLE ROUTING PROBLEM WITH TIME WINDOWS (VRPTW)

This section provides a review of the existing algorithms available in the literature to solve VRPTW and is classified as exact methods, heuristics and metaheuristics.

3.2.1 Exact Algorithms for VRPTW

Exact algorithms in the literature are based on principles of dynamic programming, lagrangean relaxation, column generation and branch-and-cut algorithm.
3.2.1.1 Dynamic Programming

Kolen et al (1987) presented the first paper on dynamic programming for the VRPTW. In this paper, the branch-and-bound approach was used in order to retrieve optimal solutions. There are three nodes in the branch-and-bound algorithm, each of which corresponds to three sets: The set of fixed feasible routes starting and finishing at the depot, a partially built route starting at the depot, and customers that are not allowed to be next on a partially built route starting at the depot. Branching is done by selecting a customer that is not forbidden and that does not appear on any route. At each branch-and-bound node, dynamic programming is used to calculate a lower bound on all feasible solutions.

3.2.1.2 Lagrangean relaxation based algorithms

Lagrangean relaxation can be applied to the VRPTW in several ways. It is well known that when the subproblem obtained by relaxing some of the constraints possesses the integrality property, the best lower bound obtained by Lagrangean relaxation is equal to the value of the linear programming relaxation of the original problem.

Fisher (1994) and Fisher et al (1997) described a Lagrangean relaxation based on m-trees. This approach relaxes the flow conservation constraints as well as the capacity and time window constraints. These constraints are relaxed in a Lagrangean fashion so that the resulting problem remains an m-tree problem with modified costs. Time windows are handled similarly by identifying infeasible paths and imposing the constraint that at least one arc in the path be left out of the solution.
In addition to the m-tree relaxation method, Fisher et al (1997) have also experimented with a variable splitting approach with additional variables introduced in the formulation, and the constraints containing additional variables are dualised. The Lagrangean sub-problem decomposes into a semi-assignment problem which is solvable by inspection, and a set of m elementary shortest path problems with time windows and capacity constraints.

Kohl and Madsen (1997) used Lagrangean relaxation to remove from the constrained region the requirement that all customers must be serviced. The resulting sub-problem turned out to be a time windows and capacity constrained shortest path problem for each vehicle that was relatively easy to solve. However, finding the optimal Lagrangean multipliers required substantial computational effort.

Kallehauge et al (2006) developed a stabilized cutting-plane algorithm to solve the Lagrangean dual. Cutting planes are generated by solving the Lagrangean subproblem and are introduced in a master problem which imposes bounds on the dual variables to ensure the stability of their values from one iteration to the next. Optimizing the relaxed master problem provides a lower bound on the value of the original problem. To obtain feasible integer solutions, the cutting-plane algorithm is embedded within a branch-and-bound algorithm and valid inequalities are introduced in the master problem. Because the relaxed master problem is stated on the dual variables, violated sub tour elimination constraints and 2-path inequalities are added as columns to this problem.
3.2.1.3 Column generation algorithms

Column generation is intimately related to constraint generation and can be seen as a special way of updating the multipliers associated with the relaxed constraints. Desrochers et al. (1992) applied column generation that was able to solve VRPTW. The basic idea is to formulate the VRPTW as a set partitioning problem by considering all feasible routes implicitly. At each major iteration a check is performed to determine there a route not currently included exists, that can reduce the cost. This is achieved by solving a relaxed shortest path sub problem with time window and capacity constraints. If such a route exists its corresponding column is added to the model and the procedure is repeated until the optimal solution for the complete set of routes is found. To obtain integer solutions the above algorithm is embedded in a branch and bound scheme.

Kohl et al (1999) developed a more efficient optimization algorithm by introducing a new valid inequality within a branch-and-cut algorithm, called k-path cuts. Due to the exponential size of the solution space, it is unlikely that these optimization procedures can be used for larger-scale problems.

Irnich and Villeneuve (2003) developed an efficient approach to forbid cycles of length greater than 2. Experiments performed by the authors show that k-cycle elimination with $k \geq 3$ can substantially improve the lower bounds.

Danna and Le Pape (2003) developed a cooperation scheme between column generation and local search applied to the VRPTW. During the branch-and-price process, local search is regularly applied from the best integer solution identified so far. This often results in an improved upper
bound that can then be used to prune nodes in the enumeration tree. Furthermore, columns associated with solutions identified during local search can be fed into the restricted master problem. The branch-and-price algorithm thus benefits from local search by being provided at an early stage with high quality upper bounds, resulting in a smaller search tree. In turn, local search benefits from branch-and-price by working with a variety of different initial solutions, resulting in an effective form of diversification.

Chabrier (2006) proposed a modified labeling algorithm to handle the constrained elementary shortest path problem and thus obtain improved lower bounds. In this algorithm, both exact and heuristic dominance rules were considered. Whenever the heuristic approach cannot find a path of negative reduced cost, the exact but slower implementation is used. This approach has allowed the author to find the optimal solution.

3.2.1.4 Branch-and-Cut Algorithm

A combination of cutting planes and exhaustive search is usually referred to as a branch-and-cut.

A branch-and-cut algorithm for the VRPTW was developed by Bard et al (2002). The algorithm incorporates five types of inequalities: subtour elimination constraints, capacity constraints, comb inequalities, incompatible pair inequalities, and incompatible path inequalities. At each node of the search tree an upper bound is computed by means of the greedy randomized adaptive search procedure (GRASP). The authors presented four separation heuristics to identify violated capacity constraints. Heuristic separation algorithms are also described for the identification of violated comb inequalities, incompatible path inequalities and incompatible pair inequalities.
Azi et al (2007) proposed an exact algorithm to solve a routing problem, where the single vehicle performs several routes over the scheduling horizon. The proposed algorithm was based on an elementary shortest path algorithm with resource constraints and it consists of two phases. In the first phase, all non-dominated feasible routes were generated by modified label correcting algorithm that solves the elementary shortest path problem with resource constraints on graphs and in the second phase, some routes were selected and sequenced to form the vehicle workday.

Calvete et al (2007) proposed a two phase approach which combines the enumeration of feasible routes and goal programming to solve optimally real medium-sized VRPTW considering hard and soft time windows. This procedure consists of first enumerating all the feasible routes and computing its deviations from targets. Second, the best sets of routes according to the goals previously established are selected by solving a set partitioning problem.

3.2.2 Heuristics

The heuristic methods for the VRPTW can be divided into construction heuristics, improvement heuristics (local searches) and metaheuristics. Construction heuristics build feasible solutions by inserting an un-routed customer into a current partial route until all the customers are routed. Improvement methods modify the current solution iteratively by performing local searches for a better solution. The metaheuristic is guided by intelligent search strategies to avoid getting trapped in local optima which result in higher-quality solutions in relatively large computational times and are well suited to solving complex problems that may be too difficult or time-consuming to be solved by traditional techniques.
3.2.2.1 Construction heuristics

Route construction methods start from an empty solution and iteratively build routes by inserting one or more customers at each iteration, until all the customers are routed. Construction algorithms are further subdivided into sequential and parallel, depending on the number of eligible routes for the insertion of a customer. Sequential methods expand at the rate of only one route at a time, whereas parallel methods consider more than one route simultaneously. Route construction algorithms are fully specified by their three main ingredients, namely, an initialization criterion, a selection criterion specifying which customers are chosen for insertion at the current iteration, and an insertion criterion to decide where to locate the chosen customers into the current routes.

3.2.2.1.1 Sequential construction algorithms

Sequential construction algorithms are mostly based on the Sweep Heuristic (Gillet and Miller 1974) and the Savings Heuristic (Clarke and Wright 1964). In the sweep heuristic, routes are constructed as an angle sweeps the location of nodes on a 2D space. In the savings heuristic, first routes are constructed in a predefined quantity and then new nodes are added to the available nodes in order to obtain maximum savings.

Baker and Schaffer (1986) proposed the first sequential construction algorithm. The algorithm is based on the savings heuristic, and starts with all possible single customer routes in the form of depot-i-depot. Then two routes with the maximum savings are combined at each iteration.

Solomon (1987) described different construction heuristics for the VRPTW: The first two are an extension of the well-known savings method
proposed by Clark and Wright (1964) and the time-oriented nearest neighbour method. Next, the author introduced three different sequential insertion methods and a time-oriented sweep method. The most successful of the three insertion heuristics is called I1, which aims to construct routes which maximize the benefit of inserting a customer into a partially constructed route, compared to serving the customer directly from the depot.

The time-oriented sweep heuristic of Solomon (1987) is based on the idea of decomposing the problem into a clustering stage and a scheduling stage. In the first phase, customers are assigned to vehicles as in the original sweep heuristic (Gillett and Miller 1974). Here a “center of gravity” is computed and the customers are partitioned according to their polar angle. In the second phase customers assigned to a vehicle are scheduled using an insertion heuristic of type I1.

Foisy and Potvin (1993) implemented the above-described parallel version of Solomon’s (1987) insertion heuristic. The parallelism is exploited in the calculation of insertion cost for each customer. The selection of the best customer for insertion is then run only on half of the available processors. To reduce the unequal workload among the processors, unrouted customers are reassigned among the processors so as to reduce the average processor’s idle time.

Balakrishnan (1993) described three heuristics for the vehicle routing problem with soft time windows (VRPSTW). The heuristics are based on the nearest-neighbor method and the Clarke-Wright savings rules, and they differ only in the way used to determine the first customer on a route and in the criteria used to identify the next customer for insertion.
Bramel and Simchi-Levi (1996) proposed an asymptotically optimal heuristic based on the idea of solving the capacitated location problem with time windows (CLPTW). In CLPTW, the objective is to select a subset of possible sites, to locate one vehicle to each site and to assign customers to the vehicles. In the VRPTW context, this selection of locations for vehicles refers to the selection of a set of seed customers that initialize the routes. The authors use a Lagrangean relaxation-based technique to solve the CLPTW and the other customers are inserted in greedy order into simple tours by favoring customers that least increase the distance traveled.

Dullaert (2000a and 2000b) argued that Solomon’s time insertion criterion underestimates the additional time needed to insert a new customer between the depot and the first customer in the partially constructed route. This can cause the insertion criterion to select suboptimal insertion places for unrouted customers. Thus, a route with a relatively small number of customers can have a larger schedule time than necessary. The author introduces new time insertion criteria to solve the problem and concludes that the new criteria offer significant cost savings.

Ioannou et al (2001) used the generic sequential insertion framework proposed by Solomon (1987) to solve a number of theoretical benchmark problems and an industrial example from the food industry. The proposed approach is based on new criteria for customer selection and insertion that are motivated by the minimization function of the greedy look-ahead solution approach of Atkinson (1994).

3.2.2.1.2 Parallel construction algorithms

Potvin and Rousseau (1993) introduced a parallel version of Solomon’s insertion heuristic I1, where the set of m routes is initialized at
once. The authors used Solomon’s (1987) sequential insertion heuristic to determine the initial number of routes and the set of seed customers. The selection of the next customer to be inserted is based on a generalized regret measure over all routes. A large regret measure means that there is a large gap between the best and second best insertion places for a customer.

Antes and Derigs (1995) proposed a parallel construction approach that constructs and improves several tours simultaneously. The approach is based on the concept of negotiation between customers and tours. First, each unrouted customer requests a service cost from every tour and sends a proposal to the tour that offered the lowest price, and second, each tour selects the most efficient proposal. The prices are calculated according to Solomon’s (1987) evaluation measures for insertion (heuristic I1). Once a feasible solution is constructed, the number of tours is reduced by one and the problem is resolved. The authors also proposed a post-optimization approach, where some of the most inefficient customers are first removed from the tours and then reinserted.

3.2.2.2 Improvement heuristics

Local search algorithms are often used to improve initial solutions generated by other heuristics. Starting from a given solution, a local search method applies simple modifications, such as arc exchanges or customer movements, to obtain neighbour solutions of possibly better cost. If a better solution is found, it then becomes the current solution and the process iterates; otherwise a local minimum has been identified.

Thompson and Psaraftis (1993) proposed a method based on the concept of cyclic k-transfers. Due to the complexity of the cyclic transfer neighborhood search, it is performed heuristically. A general methodology
developed by Thompson and Orlin (1989) is used for searching cyclic transfer neighborhoods. Savelsbergh’s 2-opt (1986) procedure is used to maintain local optimality of the routes at all times and the initial solutions are constructed using the I1 heuristic of Solomon (1987).

Russell (1995) embedded global tour improvement procedures within the tour construction process. The construction procedure used is similar to that in Potvin and Rousseau (1993). N seed points representing fictitious customers are first selected using the seed point generation procedure of Fisher and Jaikumar (1981), originally proposed for the classical VRP. The basic idea is to use vehicle capacity information to create sectors and decide the distance of the seeds from the depot within each sector. Three ordering rules are used to select the next customer for insertion, namely earliest time window, farthest distance from depot and width of the time window augmented by distance from the depot. The local search method employs a scheme developed by Christofides and Beasley (1984). In this scheme a move is performed by deleting and reinserting four customer points close to each other. For each customer, the best two routes are first determined according to the insertion cost of Solomon (1987) since it would be computationally intractable to evaluate all route assignments. This interchange procedure is applied after every k customers have been routed.

Potvin and Rousseau (1995) compared different edge exchange heuristics for VRPTW (2-opt, 3-opt and Or-opt) and introduced a new 2-opt* exchange heuristic. The basic idea in a 2-opt* is to combine two routes so that the last customers of a given route are introduced at the end of the first customers of another route, thus preserving the orientation of the routes.

Prosser and Shaw (1996) compared within-route 2-opt and inter-route operators relocate, exchange and cross for classical VRP. The
2-Opt works by reversing part of a single route. The relocate operator simply moves a customer visit from one route to another. Finally, cross is similar to the 2-opt* proposed by Potvin and Rousseau (1995) for VRPTW.

Hamacher and Moll (1996) described a heuristic for real life VRP with narrow time windows in the context of the delivery of groceries to restaurants. The algorithm is divided into two parts. In the clustering step, the customers are partitioned into regionally bounded sets using the structure of the Minimal Spanning Tree (MST). Then, customers within these sets are routed using a simple cheapest insertion algorithm followed by a local improvement, which cuts out pieces of the tour and inserts them back at another feasible location within the same tour. If a feasible solution is not found, the remaining unrouted customers are scheduled manually.

De Backer et al (1997) reported research similar to that of Prosser and Shaw (1996) in the Constraint Programming (CP) context. In CP, the computation is driven by constraints, thus giving them an active role. Looking locally at a particular constraint, the algorithm attempts to remove, from the domain of each variable involved in that constraint, values that cannot be part of any solution.

Shaw (1998) used an Limited Discrepancy Search (LDS) in the reinsertion of customers within the branch and bound procedure. The number of visits to be removed is increased during the search each time a number of consecutive attempted moves have not resulted in an improvement of the cost. LDS is used to explore the search tree in the order of an increasing number of discrepancies.

Caseau and Laburthe (1999) described a heuristic specifically designed for large routing problems. The authors introduced an LDS variation
to the parallel cheapest insertion heuristic that branches between the best and second best alternative routes for each customer if the differences in insertion costs are small. During solution construction, three moves are considered after each insertion, namely 2-opt*, reinsertion of a chain of consecutive customers from a route r into another route r’, as well as a simple customer transfer move. When no feasible insertion place can be found, three different types of moves such as swap relocate and flush and relocate are considered to make room for the unrouted customer. In cases where the number of customers on a route is less than 30, the order of the customers within the route is optimized using the exact constraint-based branch-and-bound algorithm proposed by Caseau and Laburthe (1997). Otherwise, in the case of longer routes, 3-opt is used to modify routes after each insertion. The authors also tried to restrict the customers included in each route to a particular geometric zone.

Hong and Park (1999) proposed a two-phase heuristic algorithm that consists of a parallel insertion method for clustering and a sequential linear goal programming procedure for routing. The seed customers are selected by identifying customers that cannot be served on the same route due to time or vehicle constraints. The remaining customers are inserted into these initialized tours so that the increase in route distance and waiting time is minimal. At the end of the clustering stage, groups are reformed using Or-opt and 2-opt improvement procedures. In the routing stage, the goal-programming model is decomposed into two linear programming sub problems.

Cordone and Wolfler-Calvo (2001) proposed a deterministic heuristic based on classical k-opt exchanges combined with a procedure to reduce the number of routes. The special feature of the algorithm is that it alternates between minimization of total distance and total route duration to escape from local minima. The algorithm builds a set of initial solutions using
Solomon’s insertion heuristic II, applies a local search procedure (exchanges 2 or 3 arcs) to each of them, and chooses the best one. The route reduction procedure tries to insert each customer of one route at a time into another route. If simple insertion fails, a simple ejection chain is tried, where a customer $c_j$ is first removed from the target route $r_n$ and inserted into some other route $r_m$, before inserting the current customer $c_i$ into $r_n$.

Braysy (2001a) described several local search heuristics using a new three-phase approach for the VRPTW. In the first phase, several initial solutions are created using route construction heuristics with different combinations of parameter values. In the second phase, an effort is made to reduce the number of routes using a new ejection chain-based approach (Glover 1991 and 1992) that considers reordering of the routes also. In the third phase, Or-opt exchanges are used to minimize total traveled distance. One of the construction heuristics borrowed its basic ideas from the studies of Solomon (1987) and Russell (1995). Routes are built one at a time in a sequential fashion and after $k$ customers have been inserted into the route, the route is reordered using Or-opt exchanges. In addition, new seed selection schemes are introduced. The other heuristic, draws its basic concepts from the Savings heuristic of Clarke and Wright (1964). Here, a parallel version of the Savings heuristic is implemented, and the original measure of savings is modified to consider changes in waiting times. Moreover, the customers in the combined route are reordered before evaluating the saving incurred by uniting the two routes.

Braysy (2001b) suggested two heuristics specially designed for the clustered vehicle routing problems with time windows. The first approach is similar to that of Braysy (2001a). The basic difference is the usage of four new local search operators in the third phase instead of Or-opt exchanges. These local search operators are based on modifications of the
CROSS-exchanges and the cheapest insertion heuristics. The second approach is based on identifying customers within the same cluster by forming boxes around the selected seed customers. The customers within the boxes are then ordered using their time windows, cheapest insertion heuristic and Or-opt-exchanges. In addition, if some customers are not located in any box, an attempt is made to insert them into the closest route, using the cheapest insertion heuristics. In the end, between-route customer relocations are also attempted.

### 3.2.3 Meta Heuristics

Metaheuristics are typically high-level strategies which guide an underlying, more problem-specific heuristic, to increase their performance. The main goal is to avoid the disadvantages of iterative improvement and, in particular, multiple descent by allowing the local search to escape from local optima.

#### 3.2.3.1 Genetic Algorithms

In this section we review the genetic algorithms developed for the VRPTW. Thangiah et al (1991) applied the GA to VRPTW for the first time. The GA is proposed to find good clusters of customers. The routes within each cluster are then constructed with a cheapest insertion heuristic and \( \lambda \)-interchange.

Blanton and Wainwright (1993) combined a genetic algorithm with a greedy construction heuristic. Under this scheme, the genetic algorithm searches for a good ordering of customers, while the construction of the feasible solution is handled by the greedy heuristic. The authors used the Davis encoding method, where a chromosome represents a permutation of n
customers to be partitioned into m vehicles. The evaluation function assumes that the first m customers of a chromosome are placed into the m vehicles. The remaining n-m customers are examined individually. As each new customer is selected a possible subtour is evaluated for each vehicle and the best subtour is selected. The mutation operator used randomly exchanges two genes in a chromosome. If some of the customers remain unrouted, the fitness value is the number of these unserviced customers; otherwise the fitness value is based on the total distance of the solution.

Louis et al (1999) improved the approach of Blanton and Wainwright (1993) to achieve better performance in problems with clustered customer locations. The authors built a new initialization procedure that pays attention to the clustering of customers in assigning vehicles. Here, the authors used visual clustering of customers.

Thangiah (1995a) described a cluster-first, route-second method called GIDEON that assigns customers to vehicles by partitioning the customers into sectors with a genetic algorithm. Customers within each formed sector are routed using the cheapest insertion heuristic of Golden and Stewart (1985). In the next step, the routes are improved using λ-interchanges introduced by Osman (1993). The two processes are run iteratively a finite number of times to improve the solution quality and the proposed search strategy accepts also infeasibilities during the search against certain penalty factors. The search begins by clustering customers either according to their polar coordinate angle or randomly. In the GIDEON system, each chromosome represents a set of possible clustering schemes and the fitness values are based on corresponding routing costs. The crossover operator exchanges a randomly selected portion of the bit string between the chromosomes and mutation is used with a very low probability to randomly change the bit values.
Thangiah (1995b) developed an approach GenClust similar to GIDEON. In GenClust, each chromosome encodes $n_c$ different circles, one for each cluster, and the GA is then used to search for the set of circles that leads to the best solution. Different heuristic rules are used to associate a customer with a particular cluster, when it is not contained in exactly one circle.

Potvin et al (1996) used a competitive neural network of Potvin and Robillard (1995) to select the seed customers for a modification of Solomon’s insertion heuristic (Potvin and Rousseau 1993), where several routes are constructed simultaneously. A genetic algorithm is used to find values for three constants required for this algorithm. A stochastic selection procedure is applied to the fitness values based on the number of routes and total route time of the best solution produced by the parallel insertion heuristic. A classical 2-point crossover operator is used for recombination. It swaps a segment of consecutive bits between the parents.

Benyahia and Potvin (1995) used the GA approach which is similar to that of Potvin et al. (1996) to optimize the parameter values of the sequential and parallel versions of Solomon’s (1987) insertion heuristic. The seed customers are selected as in Solomon (1987) and Potvin and Rousseau (1993) instead of using neural networks. Moreover, the authors introduced additional cost measures for insertion, involving slack and waiting times, saving of insertion compared to servicing the customer by an individual route, and the ratio of additional distance to original distance between a pair of consecutive customers.

Potvin and Bengio (1996) proposed a genetic algorithm called GENEROUS that directly applies genetic operators to solutions, thus avoiding the coding issues. The initial population is created with the Solomon’s (1987) cheapest insertion heuristic and the fitness values of the proposed approach
are based on the number of vehicles and total route time. For selection process, a linear ranking scheme is used. During the recombination phase, two parent solutions are merged into a single one, so as to guarantee the feasibility of the new solution. Two types of crossover operators are used, namely, a sequence-based one and a route-based one. A special repair operator is then applied to the offspring to generate a new feasible solution. Mutation operators are aimed at reducing the number of routes by trying to insert the customers of a randomly selected short route into other routes. Finally, in order to locally optimize the solution, a mutation operator based on Or-opt exchanges (Or 1976) is used.

Berger et al (1998) combined a genetic algorithm with construction heuristics. The initial population is created by nearest neighbor heuristic (Solomon 1987). The fitness values of the individuals are based on the number of routes and the total distance of the corresponding solution and for selection purposes the authors used roulette-wheel scheme. The proposed crossover operator combines iteratively, various routes \( r_1 \) of the parent solution \( P_1 \) with a subset of customers, formed by \( r_2 \) nearest-neighbor routes from parent solution \( P_2 \). A removal procedure is first carried out to remove some key customer nodes from \( r_1 \). Then an insertion heuristic (Solomon, 1987) coupled to a random customer acceptance procedure is locally applied to build a feasible route, considering the partial route \( r_1 \) as an initial solution. Here, only the customer nodes in routes \( r_2 \) are considered for insertion. The mutation operators are aimed at reducing the number of routes of solutions having only a few customers by trying to insert them into other routes or locally reordering routes using the Nearest Neighbor heuristic of Solomon (1987).

Braysy (1999) extended the work of Berger et al. (1998). The author proposed several new crossover and mutation operators, testing
different forms of genetic algorithms, selection schemes and scaling schemes, as well as the significance of the initial solutions. Regarding different forms of genetic algorithms, it is concluded that it is important to create many new offspring in each generation and it is enough to maintain only one population. Differences between different selection schemes are concluded to be minor. The best results were obtained with the tournament selection that performs well-known roulette-wheel scheme twice and selects the better out of the two individuals identified by the roulette-wheel scheme. A new scaling scheme, based on a weighted combination of the number of routes, total distance and waiting time, is found to perform particularly well. Finally, to create the initial population, several strategies, such as the heuristics of Solomon (1987) and randomly created routes are tried, and it is concluded that the best strategy is to create a diverse initial population that also contains some individuals with better fitness scores.

Zhu (2000) developed a genetic algorithm based on an integer representation of solutions and two new crossover operators. The initial population was obtained by a combination of solutions created by the insertion heuristic of Solomon (1987), a set of randomly created $\lambda$-interchange neighbors of the heuristic solution and totally randomly created solutions. The parents are selected with the tournament selection, and the recombination is based on selecting randomly a cut-off point in both parents and then selecting the customer right after the cut-off point from either parent. This customer is then inserted into the same position in the other parent solution, and corrections are performed to replace the duplicate customer by the replaced one. The next customer to be reinserted is selected, based either on distances or latest arrival times with respect to the previously chosen customer. Mutation is based on reversing the order of a pair or sequence of nodes. Moreover, a special hill-climbing technique is used, where a randomly selected part of the population is improved by partial $\lambda$-exchanges.
Berger et al (2001) extended the approach of Berger et al. (1998). The proposed genetic algorithm evolves two populations in parallel. The initial population is created using a random sequential insertion heuristic combined with $\lambda$-exchanges, and a reinitialization procedure based on the insertion procedure of Liu and Shen (1999). The first of the two recombination operators is the same as in Berger et al (1998). The second extends the first operator by also removing illegally routed customers and by using the insertion procedure proposed by Liu and Shen (1999) instead of Solomon’s (1987) heuristic in the reinsertion phase. Here, the also presented six mutation operator.

Tan et al (2001a) introduced a genetic algorithm similar to Zhu (2000). The representation strategy for creating the initial population and selection scheme are the same. The well-known PMX crossover operator is used to interchange gene materials between chromosomes and mutation is performed by randomly swapping nodes. The grouping of customer is determined by the insertion heuristic I1 of Solomon (1987), and the authors used $\lambda$-interchanges to create alternative groupings. After the grouping process, a hill-climbing method similar to the one used by Zhu (2000) was employed to further improve a part of the population.

Tan et al (2001b) used the messy genetic algorithm to solve the VRPTW. The solution is encoded using ordered pairs, consisting of customer and vehicle identification indexes. The initial population is generated randomly, and the construction of new individuals is performed with two operators: cut and splice and thresholding selection. Then, the routing of the customers for each vehicle is done with Solomon’s (1987) cheapest insertion heuristic. Solomon’s heuristic is also used to evaluate and check the feasibility of the solution, and create new routes for customers that cannot be included in the same route according to the current division. Thresholding
selection is then used to construct a solution by selecting and combining customer sets in the population. Here more copies are given to strings representing better solutions. A repair algorithm is then used if some customers appear several times in the solution, or some customers are missing.

Rahoual et al (2001) introduced a multicriteria genetic algorithm. Solutions are represented as integer strings, describing a chronological list of customers for each vehicle. The initial population is generated randomly and crossover is based on 2-opt* (Potvin and Rousseau, 1995). Mutation consists of random reinsertions of customers between routes and the selection is done probabilistically, based on the ranks of the individuals. Finally, the worst individuals in the population are replaced by the best ones found during the search.

Jung and Moon (2002) suggested the use of a 2D image of a solution for chromosomal cutting within a typical steady-state genetic algorithm. The initial population is created by Solomon’s (1987) I1 insertion heuristic. Fitness values are based on traveled distance, and the selection of parents for mating is performed with the binary tournament selection. Recombination is based on dividing the arcs in the selected two solutions in two sets based on different types of curves drawn on the 2D space where customers are located. Then a repair algorithm is used to include missing arcs in a nearest-neighbor manner. In mutation, each route of the offspring is split randomly, into at most three routes. After mutation, the offspring is optimized locally using three well-known improvement heuristics: Or-opt (Or 1976) and crossover and relocation of Savelsbergh (1992).

Berger and Barkaoui (2004) proposed a parallel hybrid genetic algorithm that involves the concurrent evolution of two populations of solutions wherein, the first population evolves individuals to minimize total
travelled distance and the second aims at minimizing the temporal constraint violation to generate a feasible solution.

3.2.3.2 Evolution strategies

Evolution strategies (ES) manipulate the population of individuals, which represent the solutions of an optimization problem. Due to an integrated selection mechanism the iterative calculation of a sequence of population favors the generation of better solutions. Differences to genetic algorithm exist with regard to the representation of problem and the search operators. Evolution strategies dispense with the encoding of individuals, and instead simulate the evolution process directly on the level of problem solutions. In contrast to the genetic algorithm, mutation operators are given a superior role in comparison to the recombination operators.

Homberger and Gehring (1999) proposed two evolution strategies for the VRPTW. The individuals of a starting population are generated by means of a stochastic approach, which is based on the savings algorithm of Clarke and Wright (1964). In this approach, the stochastic element consists of the random selection of savings elements from the savings list. Selection of the parents is done randomly and only one offspring is created through the recombination of parents. In this way, a number $\lambda > \mu$ offspring is created, where $\mu$ denotes the population size. At the end, fitness values are used to select $\mu$ offspring for the next population. The fitness values are based on the number of routes, total travel distance and a criterion that determines how easily the shortest route of the solution can be eliminated. The mutation is based on local searches of Or-opt (Or, 1976), 2-opt* (Potvin and Rousseau 1995) and $\lambda$–interchange-move (Osman 1993) with $\lambda = 1$. In addition, a special Or-opt based operator is used to reduce the number of routes. The first
out of the two proposed metaheuristics, evolution strategy ES1, skips the recombination phase. The second evolution strategy ES2 uses the uniform order-based crossover (Goldberg 1989) to modify the initially randomly created mutation codes. The mutation code is used to control a set of removal and insertion operators performed by the Or-opt operator. The strategy parameters refer to how often a randomly selected local search operator is applied, and to a binary parameter used to alternate the search.

Mester (2002) has also experimented with evolution strategies similar to Homberger and Gehring (1999). In the beginning, all customers are served by separate routes. Then, a set of six initial solutions is created using the cheapest reinsertions of single customers with varying insertion criteria. The best initial solution obtained is used as a starting point for the ES. The multi-parametric mutation consists of removing a set of customers from a solution randomly, based on the distance to the depot or by selecting one customer from each route. Then, a cheapest insertion heuristic is used to reschedule the removed customers. After mutation, the solution vector is improved using the same three improvement heuristics as in Homberger and Gehring (1999) described above, and, in addition, with the GENIUS heuristic of Gendreau et al (1992).

3.2.3.3 Tabu Search

Tabu search (TS) presented by Glover (1986) is a memory based local search heuristic. In TS, the solution space is searched by moving from a solution s to the best solution in its neighborhood N(s) at each iteration. In order to avoid local optimum, the procedure does not terminate at the first local optimum and the solution may be deteriorated at the following iteration. The best solution in the neighborhood is selected as the new solution even if it is poorer. Solutions having the same attributes with the previously searched
solutions are put into the tabu list and moving to these solutions is forbidden. This usually prevents making a move to solutions obtained in the last t iterations. TS can be terminated after a constant number of iterations without any improvement of the overall best solution or a constant number of iterations.

Garcia et al (1994) described a tabu search heuristic where the neighborhood is restricted to the exchange of arcs that are close in distance. The initial solution is created using Solomon’s I1 insertion heuristic, and the algorithm oscillates between the 2-opt* (Potvin and Rousseau 1995) and Or-opt (Or 1976) exchanges. When one has not made any improvement for a certain number of iterations, the other improvement operator is used and vice versa. In order to minimize the number of routes, the algorithm tries to move customers from routes with a few customers into other routes using the Or-opt exchanges. The parallel implementation is performed by partitioning the neighborhood among slave processors. The master processor is then used to guide the tabu search. After exploration of the neighborhood, the best move from each processor is sent to the master.

Rochat and Taillard (1995) proposed a tabu search approach based on adaptive memory and 2-Opt local search. In the first phase, tabu search is used to create a number of different solutions and these solutions are then stored in the adaptive memory. The selection of routes from memory is done probabilistically and the probability of selecting a particular route depends on the value of the solution the route belongs to. The selected tours are improved using tabu search and inserted subsequently back into adaptive memory. At the end, a set partitioning problem is solved exactly, using the routes in the pool to create the best possible solution.
Carlton (1995) described a reactive tabu search that dynamically adjusts its parameter values based on the current search status. More precisely, the size of the tabu list is managed by increasing the tabu list size if identical solutions occur too often and reducing it if no feasible solution can be found. This approach is applied to several types of problems with time windows. Its robustness comes from a simple neighborhood structure, which can be easily adapted to different problems.

Bachem et al (1996) described an improvement heuristic based on the mechanisms of trading. The partition of customers into the tours is determined by finding matches in a trading graph. The nodes correspond to either an insertion (buy) of a customer into a tour or a deletion (sell). The edges represent possible exchanges and the weight of each edge is the gain that is obtained by the corresponding action. Thus, every matching of the trading graph corresponds to a number of interchanges of customers. In each iteration, tours were shuffled by choosing some permutation at random. Then, for each tour either a sell or buy action is selected and finally possible trading matches are evaluated and the best one selected. This approach allows infeasibilities against certain penalty factors, as well as trading matchings with negative weights causing deterioration. Because of this deterioration a tabu list is also added to prevent cycling.

Taillard et al (1997) proposed a tabu search heuristic for the vehicle routing problem with soft time windows. The authors proposed a new exchange heuristic called CROSS exchange which is used both to exchange customers between routes and for intra-route optimization. Solomon’s insertion heuristic I1 with random parameters is first used to fill the adaptive memory (Rochat and Taillard 1995) with different types of routes. The solution is then decomposed into a disjoint subset of routes by using the polar angle associated with the center of gravity of each route, and tabu search is
used to solve each subset separately. A complete solution is reconstructed by merging the new routes found by tabu search. Decomposition, tabu search and reconstruction are repeated for a certain number of iterations. The algorithm penalizes frequently performed exchanges to diversify the search and reorders the customers within the best routes using Solomon’s II insertion heuristic. Moreover, an adaptation of the GENIUS heuristic (Gendreau et al 1992) for time windows is applied to each individual route of the final solution.

Badeau et al (1997) proposed a framework similar to that of Taillard et al (1997), using a 2-level parallel implementation that combines the so-called master-slave scheme with an allocation of each subproblem to a different processor. In this master-slave scheme, the master process manages the adaptive memory and generates solutions from it; these solutions are then transmitted to slave processes that improve them by performing tabu search, and return the best solutions found, to the master.

Chiang and Russell (1997) developed a reactive tabu search that dynamically varies the size of the list of forbidden moves to avoid cycles as well as an overly constrained search path. The underlying local search is based on the $\lambda$ interchange mechanism of Osman (1993) with $\lambda = 1$. The tabu search is applied to the parallel construction approach of Russell (1995) that incorporates improvement procedures during the construction process and builds several routes simultaneously. The frequency-based diversification strategy penalizes the customer nodes who switch too frequently. The intensification strategy is designed to reduce the waiting time at each customer. This is specifically achieved by forbidding certain customers from moving into another route.

De Backer and Furnon (1997) described a classical two-phase mechanism to solve TSP, VRP and VRPTW. The initial solution is first
generated using the savings heuristic of Clarke and Wright (1964). This solution is subsequently optimized using two intra-route local searches (2-Opt and Or-opt) and three inter-route operators (cross, exchange and relocate) guided with tabu search. Here, recently removed or inserted arcs are unauthorized, i.e., tabu for a given amount of time. The approach is coupled to a Constraint Programming framework, where feasibility of the new solution is checked through constraint propagation.

Brandao (1999) described a tabu search algorithm allowing infeasible solutions during the search process. The initial solution is created with a simple sequential cheapest insertion algorithm. Here, only unrouted customers that are close to customers already inserted into the route are considered for insertion. The created solution is improved by inserting randomly selected customers into other routes or into another place within the same route, or by swapping randomly selected customers between two routes. After each successful insertion or swap, the routes with a new customer are reordered using a modified version of the GENI algorithm of Gendreau et al (1992).

Schulze and Fahle (1999) proposed a tabu search performing several search threads in parallel. Each thread is started with a different initial solution and a neighboring solution is generated by performing a sequence of simple customer shifts (ejection chain). All routes generated by the tabu search heuristic are collected in a pool. At the termination of local optimization steps, the worst solution is replaced by a new one created by solving the set covering problem on the routes in the pool with Lagrangian relaxation. With this new set of solutions, the whole process is restarted until a certain stopping criterion is fulfilled.
In addition, the proposed method tries to eliminate routes having at the most three customers by trying to move these customers into other routes. The routing of customers supplied by the same vehicle is improved by performing Or-opt exchanges within the route and the search is diversified by penalizing frequently performed customer shifts. To generate an appropriate number of initial solutions, three different heuristics are used, namely Solomon’s (1987) I1 insertion heuristic, the parallel route building heuristic of Potvin and Rousseau (1993) and a modified version of the Savings heuristic of Clarke and Wright (1964). In the parallel implementation, each processor handles a set of solutions instead of just one and also solves the set covering problem separately on these solutions to avoid idle times. Each time a processor terminates its local optimization process, the routes of the optimized solutions are sent to all other processors to enable sharing of knowledge.

Gehring and Homberger (1999 and 2001) developed a two-phase approach, where the tabu search is combined with the evolutionary algorithm ES1 of Homberger and Gehring (1999). In this evolutionary algorithm the search is mainly driven by mutation based on the Or-opt (Or, 1976), 2-opt* (Potvin and Rousseau, 1995) and λ interchange moves (Osman 1993) with λ = 1. In addition, a special Or-opt-based operator is used to reduce the number of routes. The individuals of a starting population are generated by means of a stochastic approach that is based on the savings algorithm of Clarke and Wright (1964). The evolutionary algorithm is used in the first phase to minimize the number of routes. In the second phase, the total distance is minimized using a tabu search algorithm utilizing the same local search operators. The approach is parallelized using the concept of cooperative autonomy.
Tan et al (2000) explained a tabu search algorithm based on \( \lambda \) interchanges with best-accept strategy. The initial solution is created with cheapest insertion heuristics by Thangiah et al (1994). Each time a local minimum is found, the search is diversified by performing a series of random \( \lambda \) interchange hops combined with the 2-opt* operator. A candidate list is maintained to record elite solutions discovered during the search process. These elite solutions are then used as a starting point for intensification.

Lau et al (2001) presented a generic constraint-based diversification technique to enhance a tabu search algorithm for VRPTW. An initial solution is first created with a simple greedy insertion algorithm. Then a local optimal solution is generated by tabu search using exchange and relocate operators with the best-accept strategy. Using a strategy similar to Large Neighborhood Search by Shaw (1998), a part of this solution is extracted and passed to the constraint-based local search.

### 3.2.3.4 Simulated Annealing

Simulated Annealing (SA) is a stochastic relaxation technique. It is based on the annealing process of solids, where a solid is heated to a high temperature and gradually cooled in order to crystallise it. During the SA search process, the temperature is gradually lowered. At each step of the process, a new state of the system is reached. If the energy of the new state is lower than the current state, the new solution is accepted. But if the energy of the new state is higher, it is accepted with a certain probability. This probability is determined by the temperature. SA continues searching the set of all possible solutions until a stopping criterion is reached.

Chiang and Russell (1996) proposed three different SA methods. The first one uses a modified version of the k-node interchange mechanism
and the second uses the $\lambda$-interchange with $\lambda=1$. The third is based on the concept of the tabu list of Tabu Search.

Thangiah et al (1999) used $\lambda$-interchange with $\lambda=2$ to define the neighborhood and decrease the temperature after each iteration. In case the entire neighborhood has been explored without finding and accepting moves, the temperature is increased.

Tan et al (2001c) proposed an SA heuristic. They defined a new cooling schedule. Thus, when the temperature is high, the probability of accepting the worse is high, when the temperature is decreased according to the new cooling schedule, the probability of accepting the worse is reduced.

Li and Lim (2003) proposed an algorithm that finds an initial solution using Solomon’s insertion heuristic. Local search is then performed from this solution using three exchange operators that move segments of customers either between routes or within the same route. Whenever a local minimum is reached, multiple restarts are performed starting from the best known solution using the simulated annealing approach, and a tabu list is used to prevent cycling.

Bent and Van Hentenryck (2004) described a two-stage hybrid algorithm. The first stage uses a lexicographic evaluation. The neighbourhood used in this stage consists of 2-opt, Or-opt, relocating, exchange and crossover moves. In the second stage, subsets of customers are removed from their current route and reinserted in possibly different routes. Customers selected for removal are randomly chosen but the algorithm favors customers that are geographically close to each other and belong to different routes. A branch-and-bound algorithm is then used to reinsert these customers.
3.2.3.5 Ant Colony Optimisation

Gambardella et al (1999) presented a Multiple Ant Colony System for the Vehicle Routing Problem with Time Windows (MACS-VRPTW), an ACO based approach for solving the VRPTW. MACS-VRPTW has a multiple objective function and both objectives are optimized simultaneously by the coordination of two ant colonies.

Ellabib et al (2002) proposed another Ant System (AS) based approach for solving the VRPTW. The basic idea is to let the Ant Colony Optimisation (ACS) perform its search in the space of local minima rather than in the search space of all feasible tours. The VRPTW is transformed to the TSP as proposed by Gambardella et al (1999). The approach starts by applying a tour construction heuristic for creating a good initial solution and then lets the ACS operate on the search space of local optima to guide the search toward the global optimum.

Reimann et al (2002) proposed ant system based approach to the VRPBTW. In this algorithm an insertion procedure based on Solomon (1987) is used to construct solutions. The routes are constructed one by one. First, the furthest customer from the depot is selected as the seed customer for the current route. Sequentially other customers are inserted into this route until no more insertions are feasible. After routes are constructed, swap and move procedures are applied to improve the solution. In this method only global pheromone updating is applied, and the pheromone trails are updated.

Chen and Ting al (2005) proposed a new hybrid algorithm (IACS-SA) that combines an improved ant colony system (ACS) with simulated annealing (SA). The improved ant colony system (IACS) possessed a new construction rule, a new pheromone update rule and diverse local
search approaches (2-opt and Insertion Move). The new hybrid algorithm combines the strengths of both search heuristics. In IACS-SA, IACS can provide a good initial solution to SA and SA can assist IACS to escape from local optima.

Ellabib et al (2007) developed multiple ant colony system with new exchange strategies based on a weighting scheme for VRPTW. In the proposed strategy, the authors adjusted the pheromone matrix of each colony through different colony level interaction and according to the solution information. Through weighting scheme, the pheromone trails on the edges of the best solutions are updated adoptively in response to determined weights and an extra amount of pheromone is deposited on the edges of solutions accordingly.

3.2.3.6 Hybrid Methods

In this section, heuristic search methods that hybridize ideas of evolutionary computation with some other search techniques are reviewed.

Brasy et al (2000) hybridized a genetic algorithm and an evolutionary algorithm. In the first phase, a genetic algorithm based on Berger et al (1998) and Brasy (1999) is used to obtain a feasible solution. The evolutionary algorithm used in the second phase picks every combination of two routes in random order and applies randomly one out of the four local search operators or route construction heuristics. Offspring routes generated by these crossover operators are mutated according to a user-defined probability by selecting randomly one out of two new local search operators. Selecting each possible pair of routes, mating and mutation operators are repeatedly applied to a certain number of generations and finally a feasible
solution is returned. To escape from a local minimum, arcs longer than average are penalized if they appear frequently during the search.

Wee Kit et al (2001) described a hybrid genetic algorithm, where a simple tabu search based on cross, exchange, relocate and 2-Opt neighborhoods is applied to individual solutions in the later generations to intensify the search. The genetic algorithm used is based on a random selection of parent solutions and two new crossover operators. The first operator tries to modify the order of the customers in the first parent by trying to create consecutive pairs of customers according to the second parent. The second crossover operator tries to copy the common characteristics of parent solutions to the offspring, by modifying the seed selection procedure and cost function of an insertion heuristic similar to that of Solomon (1987).

Braysy (2002) presented a three phase algorithm for vehicle routing and scheduling problems with time windows. In the first phase, the initial seed is generated by hybrid construction heuristic and merge heuristic. The second phase uses local search operator based on ejection chain to reduce the number of routes and in the third phase Or-opt exchange is applied to minimize the distance.

Braysy and Dullaert (2003) extended the work of Braysy et al (2000). Here the genetic algorithm of the first phase is replaced by a two-stage multi-start local search (Yagiura and Ibaraki 2001). In the first stage, a set of initial solutions is created using a sequential cheapest insertion heuristic of Braysy (2003). After creating an initial solution, an attempt is made to reduce the number of routes, using a special ejection chain. The main innovation of this approach is a reordering of the routes within the ejection chain to reduce the lateness. In the second phase, an evolutionary algorithm is used to minimize the total distance. The differences with respect to the
previous study lie in the number and type of crossover and mutation operators used. Here, just two crossover operators are used. Finally, the offspring routes generated by the crossover operators are mutated using modified intra-route CROSS-exchanges.

Berger et al (2003) proposed an algorithm that relies on the concept of a simultaneous evolution of two populations pursuing different objectives subject to partial constraint relaxation. Genetic operators have been designed to incorporate key concepts emerging from recent promising techniques such as insertion heuristics and large neighborhood search to explore the solution space.

Braysy (2003) developed a new deterministic metaheuristic based on a modification of the variable neighborhood search of Hansen and Mladenovic (2001). The proposed procedure is based on a new four-phase approach. In this approach an initial solution is first created using new route construction heuristics followed by a route elimination procedure to improve solution the regarding the number of vehicles by using the new ejection chain approach. In the third phase, solutions are improved in terms of total distance using VNS oscillating between four new local search procedures. Finally, in the fourth phase the best solution obtained is improved by modifying the objective function to escape from a local minimum.

Braysy et al. (2004) presented a two-phase multi- start local search for the VRPTW. In the first phase, a sequential insertion heuristic is used to generate a number of initial solutions and all solutions are improved by a tour depletion approach. In the second phase, all solutions with the minimal number of routes are further improved with respect to distance by intra- and inter-tour operators. The best solution obtained after second phase were post-optimized by a threshold accepting post-processor.
Bent and Hentenryck (2004) proposed a two stage hybrid algorithm. The algorithm first minimizes the number of vehicles using simulated annealing. It then minimizes the travel cost using a large neighborhood search which may relocate a large number of customers.

Ioannou and Kritikos (2004) used a three-stage method for solving VRPTW. In the first stage, using the Hungarian method, the optimal customer matching for an assignment approximation based on travel time relaxation is obtained. In the second stage, the assignment matching is transformed into feasible routes by means of a simple decoupling heuristic. In the last stage, the best of these routes found based on travelling and waiting times constitute partial route of the final solution, which is then completed by the routes provided by heuristic methods applied to the remainder of the customers.

Ibaraki et al (2005) proposed local search algorithms for the vehicle routing problem with soft time window constraints. In the algorithm, local search is used to assign customers to vehicles and to find orders of customers for vehicles to visit. It employs an advanced neighborhood, called the cyclic exchange neighborhood, in addition to standard neighborhoods for the vehicle routing problem. After fixing the order of customers for a vehicle to visit, we must determine the optimal start times of processing customers so that the total penalty is minimized by using dynamic programming.

Le Bouthillier et al (2005) presented a parallel co-operative methodology in which several agents communicate through a pool of feasible solutions. The agents consist of simple construction and local search algorithms to initialize the pool. The evolutionary algorithms use a probabilistic mutation and the well-known edge recombination and order crossovers, while the tabu search procedures are adaptations of the
TABUROUTE method of Gendreau et al (1994) and allow for infeasible solutions during the search.

Mester and Braysy (2005) hybridized the evolution strategies of Mester (2002) with the Guided local search metaheuristic. It is based on an iterative two-stage procedure, where Guided Local search is used to regulate a composite local search in the first stage, and the objective function and neighborhood of the modified evolutionary strategies local search algorithms of Mester (2002) in the second stage. The two stages are repeated iteratively until the stopping criterion is met. The composite local search is based on the well-known relocate, 1-intercange and 2-opt* neighborhoods and the initial solution is created with the cheapest insertion heuristic of Mester (2002).

Le Bouthillier and Crainic (2005) proposed a parallel cooperative methodology in which several agents communicate through a pool of feasible solutions. The agents consist of simple construction and local search algorithms and four different metaheuristic methods, namely, two evolutionary algorithms and two tabu searches. The evolutionary algorithm uses a probabilistic mutation and the well-known edge combination and order crossovers, while tabu search procedures are adaptations of the TABUROUTE method of Gendreau et al (1994) and the unified tabu search of Cordeau et al (2001).

Le Bouthillier et al (2005) developed a pattern identification mechanism that endows cooperative search with capabilities to create new information, and guide the global search. The proposed mechanism sends information to independent metaheuristics about promising and unpromising patterns in the solution space. By fixing or prohibiting specific solution attribute values in certain search metaheuristics, we can focus the search on desired regions. The mechanism thus enforces better coordination between
individual methods and controls the global search’s diversification and intensification.

Csiszar (2005) proposed a two phase approach. In the first phase, the main focus is on, route elimination. In the second phase, the focus is on cost reduction. For this, a new model called “Magic Bricks” is proposed. The model suggests, let the width of a brick be the distance between two nodes on any route and the waiting time is the gap between the bricks. Similarly, a single route can be considered a row of bricks in the wall and the whole number of routes would create a wall. The objective of VRP can be redrafted: rebuild the wall to get a primarily smaller wall - with fewer routes – secondly, try to reduce the length of the brick walls.

Alvarenga et al (2007) proposed a three phase approach. Initially, a hierarchical tournament selection genetic algorithm is applied. Then, the two phase approach, the genetic and set partitioning, is applied. The Push Forward Insertion Heuristic (PFIH) (Solomon 1987) is used to generate the initial population. Nine fitness criteria are defined to permit the identification and to eliminate one more route or a customer in the shortest route. A new set of mutations are defined.

Sontrop et al (2005) introduced new ejection chain strategies. Ejection chain procedures are based on the idea of compound moves that allow a variable number of solution components to be modified within any single iteration of a local search algorithm. The yardstick behind such procedures is the underlying reference structure, which is used to coordinate the moves that are available for the local search algorithm. A new reference structure is proposed, which is a generalization of the doubly rooted reference structure, resulting in a new powerful neighborhood for the VRPTW. Tabu search is used for the generation of ejection chains. On a higher algorithmic
level, the effect of different metaheuristics such as iterated local search and simulated annealing to steer the tabu chain ejection process is studied.

Pisinger and Ropke (2007) presented a new approach called Adaptive Large Neighborhood Search (ALNS), an extension of LNS with an adaptive layer. This layer adaptively chooses a number of insertion and removal heuristics, to intensify and diversify the search. In each iteration the solution is chosen to destroy the current solution, and an algorithm is chosen to repair the solution. The new solution is accepted if it satisfies some criteria defined by the local search framework applied at the master level. It first transforms a VRPTW instance to a rich pickup and delivery problem with time windows (RPDPTW) and then solves it using ALNS.

3.3 MULTI-DEPOT VEHICLE ROUTING PROBLEM WITH TIME WINDOWS (MDVRPTW)

MDVRPTW is an NP hard problem (Bodin et al 1983). It is a variant of the VRPTW, considering multiple depots at which the vehicles are based. Even though, intensive research publications are available for both heuristic and exact optimization approaches in the area of VRP and VRPTW, very limited study has been carried out in the area of MDVRPTW.

Cordeau et al (2001) have presented a unified Tabu search heuristic to solve the routing problem with time windows (VRPTW) and two of its variants: MDVRPTW and Period vehicle routing problem (PVRPTW). The main feature of this approach has been the use of very simple neighborhood, a move of one customer from one route to another. Infeasible solutions have been allowed during the search and to ensure that the algorithm moves from infeasible to feasible area of search space, an adaptive penalty function has been used.
Cordeau et al (2004) have given an improved Tabu Search algorithm considering the route duration constraint. They have addressed the improved version of the neighborhood search mechanism of their earlier work.

Polacek et al (2004) have applied the philosophy of the Variable neighborhood search (VNS) proposed by Mladenovic and Hansen (2001). The initial solution for the MDVRPTW is constructed by assigning each customer to the nearest depot. Then all customers within a depot are ordered by the center of their time window and the routes are constructed for each depot. CROSS-exchange (Taillard et al 1997) and i CROSS-exchange (Braysy 2003) operators are used to define a set of neighborhood structures to find a balance between effectiveness and the chance to get out of local optima in the shaking phase of VNS. The solution obtained through shaking is afterwards submitted to a local search to come up with a local optimal solution. The local search mechanism adopted by them is a restricted version of the 3-Opt. After the shaking and the local search procedures have been performed, the solution obtained was compared with the incumbent solution to decide whether or not to accept it.

Chiu et al (2006) presented a two phase heuristic method with an embedded diverse and greedy strategy. The Phase one heuristic uses a modified version of Solomon’s (1987) construction method that simultaneously considers clusters and routes and is adopted to generate several initial solutions. In phase two, to improve the initial solutions, two procedures, namely, inter-route and intra-route improvement procedures are used.

Dondo and Cerda (2007) proposed a hierarchical hybrid solution approach that integrates a heuristic clustering procedure into an optimization framework for the multi-depot heterogeneous fleet Vehicle Routing Problem
with Time Windows. In this work they have developed a three phase hierarchical hybrid approach in which the clusters of nodes are first defined, then such clusters are assigned to vehicles and sequenced on the related tours, and finally the routing and scheduling for each individual tour is found.

3.4 RESEARCH GAP

- Algorithms for VRPTW and MDVRPTW are considerably more intricate than classical VRP and it calls for efficient algorithm so that immediate requests can be served. These have made them an important candidate for solution using metaheuristics.

- Even though there are many literatures dealing with the application of metaheuristics for solving the VRPTW, not much work is reported on the application of metaheuristics for MDVRPTW.

- In most of the existing literature, researchers consider either minimizing the number of vehicles or minimizing the distance traveled, as their objective, while solving SDVRPTW and MDVRPTW. But, in the proposed method an attempt has been made to minimize both the vehicles used as well as the distance travelled.
3.5 SCOPE OF THE STUDY

The scope of this research work is to develop metaheuristics based search methods to solve VRPTW and MDVRPTW and extending these metaheuristics for designing vehicle routes for the case of third party logistic service provider.

3.6 SUMMARY

A comprehensive review of the existing exact methods, heuristics and metaheuristics for Single Depot and Multi-Depot Vehicle Routing Problem with Time windows has been presented in this chapter. The research gap and scope of the study has been indicated.