Algorithmic Operations Research

Vehicle Routing Problem with Time Windows:
An Evolutionary Algorithmic Approach

Saswati Tripathi and Bhawna Minocha

Volume 1, Number 2, Summer 2006

URI: https://id.erudit.org/iderudit/aor1_2art05

See table of contents

Publisher(s)

Preeminent Academic Facets Inc.

ISSN

1718-3235 (digital)

Explore this journal

Cite this article

Tripathi, S. & Minocha, B. (2006). Vehicle Routing Problem with Time Windows:: An Evolutionary Algorithmic Approach. *Algorithmic Operations Research*, *1*(2), 104–118.

Article abstract

The Vehicle Routing Problem with Time Windows (VRPTW) is an important problem in logistics, which is an extension of well known Vehicle Routing Problem (VRP), with a central depot. The Objective is to design an optimal set of routes for serving a number of customers without violating the customer's time window constraints and vehicle capacity constraint. It has received considerable attention in recent years. This paper reviews the research on Evolutionary Algorithms for VRPTW. The main types of evolutionary algorithms for the VRPTW are Genetic Algorithms and Evolutionary Strategies which may also be described as Evolutionary metaheuristics to distinguish them from other metaheuristics. Along with these evolutionary metaheuristics, this paper reviews heuristic search methods that hybridize ideas of evolutionary algorithms with some other search technique, such as tabu search, guided local search, route construction heuristics, ejection chain approach, adaptive large neighborhood search, variable neighborhood search and hierarchal tournament selection. In addition to the basic features of each method, experimental results for the 56 benchmark problem with 100 customers of Solomon (1987) and Gehring and Homberger (1999) are presented and analyzed.

Algorithmic Operations Research Vol.1 (2006) 104–118

Vehicle Routing Problem with Time Windows: An Evolutionary Algorithmic Approach

Saswati Tripathi [a] Bhawna Minocha

[a] Amity School of Computer Sciences, Noida, India

Abstract

The Vehicle Routing Problem with Time Windows (VRPTW) is an important problem in logistics, which is an extension of well known Vehicle Routing Problem (VRP), with a central depot. The Objective is to design an optimal set of routes for serving a number of customers without violating the customer's time window constraints and vehicle capacity constraint. It has received considerable attention in recent years. This paper reviews the research on Evolutionary Algorithms for VRPTW. The main types of evolutionary algorithms for the VRPTW are Genetic Algorithms and Evolutionary Strategies which may also be described as Evolutionary metaheuristics to distinguish them from other metaheuristics. Along with these evolutionary metaheuristics, this paper reviews heuristic search methods that hybridize ideas of evolutionary algorithms with some other search technique, such as tabu search, guided local search, route construction heuristics, ejection chain approach, adaptive large neighborhood search, variable neighborhood search and hierarchal tournament selection. In addition to the basic features of each method, experimental results for the 56 benchmark problem with 100 customers of Solomon (1987) and Gehring and Homberger (1999) are presented and analyzed.

Key words: Vehicle Routing Problem with Time Windows, Evolutionary Algorithms, Genetic Algorithms and Evolutionary Strategies

1. Introduction

The Vehicle Routing Problem (VRP) is an NP-hard and very well known combinatorial optimization problem which finds many practical applications in the design and management of distribution systems. It can be described as follows: given a set of vehicles with uniform capacity, a common depot and several customer demands (represented as a collection of geographical scattered points), the objective of the VRP is to design minimum cost routes, by visiting each point exactly once, with the restriction that all routes start and end at the depot and the total demand of all points on one particular route can not exceed the capacity of the vehicle.

Vehicle Routing Problem with Time Windows (VRPTW) is one of the most renowned problems in contemporary operations research. VRPTW is the generalization of the VRP where the service at each customer must start within an associated time windows and the vehicle must remain at the customer location during service. Soft time windows can be violated at a cost, while hard time windows do not allow for a vehicle to arrive at a customer after the latest time to begin service. If it arrives, before the customer is ready to begin service, it waits.

It can be described as follows. Let G= (V, E) be a connected directed graph with node set V = V_NU $\{v_0\}$ and arc set E, where V_N = $\{$ v_i V | i= 1, 2,. . . ,n$\}$ stands for customers, each of which can be serviced only within a specified time interval and $v_0$stands for the central depot, where all routes start and end. Each node v_i V has an associated demand q_i that can be a delivery form or a pickup for the depot and service time s_i with service window [e_i,l_i]. The set E of arcs with non negative weights represents the travel distances d_{ij} between every two distinct nodes $v_i$and $v_j$and the corresponding travel time t_{ij}. If the vehicle reaches the customer $v_i$before the e_i, a waiting time occurs. The routes schedule time is the sum of the travel time, waiting time and the service time. The primary objective of VRPTW is to find the minimum numbers of tours, without violating vehicle's capacity constraints Q and the customer's time windows. The tours correspond to feasible routes starting and ending at the depot. A secondary objective is often to minimize the total distance traveled or to minimize total schedule time. All problem parameters, such as customer demands, travel times and time windows

Email: Saswati Tripathi [stripathi@ascs.amity.edu], Bhawna Minocha [bminocha@ascs.amity.edu].

are assumed to be known with certainty.

VRPTW is NP-hard. Even finding a feasible solution to the VRPTW with a fixed vehicle size is itself an NP-hard problem. Techniques like exact optimization and heuristics approaches are used for solving VRPTW. Main focus in early surveys of solution techniques for the VRPTW, Fisher [21], Toth et al [57], are on exact optimization techniques. Because of the complex nature of VRPTW and its wide applicability to real life situations, solution techniques like heuristic which are capable of producing high quality solutions in limited time are of prime importance. But, over the last few years Metaheursitics are the core of recent work on methods for the VRPTW. Unlike local search heuristic that terminates once a local optima has been found, these methods explore a larger subset of the solution space in the hope of finding a near optimal solution.

Metaheuristics such as tabu search (Cordeau et al [15]), ant algorithms (Gambardella [24]), simulated annealing (Thangiah et al [55]) and evolutionary algorithms (Hömberger et al [31]) are used as solution techniques for VRPTW. In this paper we surveyed evolutionary algorithms and compared them with other metaheuristics both basic such as tabu search, ant colony system and hybrid algorithms, for VRPTW. Section 2 divides evolutionary algorithms developed for VRPTW into genetic algorithms and evolutionary strategies. Section 3, discusses hybridize ideas of evolutionary computation with some other search technique, such as tabu search, guided local search, route construction heuristics, ejection chain approach, adaptive large neighborhood search, variable neighborhood search and hierarchal tournament selection followed by section 4 which contains other important hybrid algorithms. In section 5, we described computational results for the some of the described metaheuristics and Section 6 concludes the paper.

2. Evolutionary Algorithms

Evolution is a phenomenon of adapting to the environment and passing on genetic information to following generations. The first algorithms that use a natural evolution as the central strategy to solve problems were published in the 50s, such as Fraser [23] and Box [5]. In 1966 Forgel et al. [22] proposed a method called Evolutionary Programming. Following that, in 1973, Rochenberg [47] introduced the method called Evolution Strategies. The proper Genetic Algorithm, or simply GA, was proposed by Holland [30] in 1975. All these proposals were based in the natural reproduction, selection and evolution theory from Darwin [18].

Evolutionary algorithms or EA is characterized by maintaining asset of solution candidates that undergoes a selection process and is manipulated by genetic operators. By analogy to natural evolution, the solution candidates are called individuals and the set of solution candidates is called the population. Each individual represents a possible solution to the problem at hand. An individual does not act as a decision vector but rather encodes the solution to the optimization problem into a decision vector based on an appropriate structure, e.g., a bit vector or a real-valued vector. Each subsection of the data structure holding the encoded solution (chromosome) is called a gene, and it usually encodes the value of a single parameter.

Selection is a process in which design candidates (parents) are selected for recombination based on their fitness values. Fitness refers to measure of profit, utility or goodness to be maximized while exploring the solution space. Recombination (or crossover) and mutation are genetic operators aiming at generating new solutions within the search space from existing ones. The crossover operator combines information from a certain number of parents to create a certain number of children (offspring). The mutation operator modifies individuals by randomly changing (typically) small parts in the associated decision vectors according to a given probability (mutation rate). Both crossover and mutation work on individuals, not on the decoded decision vectors.

Based on the above concepts, natural evolution is simulated by an iterative computation process. In the beginning a population of candidate solutions to a problem at hand is initialized. This is often accomplished by randomly sampling from the solution space. Then a loop consisting of parent evaluation (fitness assignment), selection, recombination and/or mutation is executed a certain number of times. Each loop iteration is called a generation, and the search is stopped once some convergence criteria or conditions are met. Such criteria might, for instance, refer to a maximum number of generations or the convergence to a homogeneous population composed of similar individuals. They are divided into three main subclasses: genetic algorithms (GA), evolution strategies (ES) and evolutionary programming (EP).

Evolutionary programming, originally conceived by L. Forgel [22], represents individuals phenotypically as finite state machines capable of responding to environmental stimuli, and developing operators (primarily mutation) for effecting structural and behavioral change over time. EP was later reintroduced in the early 1990s. The new evolutionary programming is nearly identical to evolution strategies, using similar mutation strategies and a slightly different selection process.

These early characterizations, however, are no longer useful in describing the enormous variety of developed evolutionary algorithms. The literature is filled with new terms and ideas such as memetic algorithms that combine local search with recombination. Memetic Algorithms is a population-based approach for heuristic search in optimization problems. For some problems they have been shown to be more efficient than genetic algorithms. Some researchers view them as hybrid genetic algorithms or parallel genetic algorithms.

Most of the evolutionary methods developed for the VRPTW are hybrids, incorporating real-value representation and a set of construction heuristics and local searches. Nevertheless, the authors call them genetic algorithms. As a consequence, labels such as genetic algorithm are not that helpful in understanding the algorithm in question. In this paper, we try to avoid this problem by focusing on basic elements common to all evolutionary algorithms, and using them to understand the differences and analyze the genetic algorithms, evolution strategies, hybrid algorithms with genetic or evolutionary components developed for the VRPTW.

2.1. *Genetic Algorithms*

The proper Genetic Algorithm, or simply GA, was proposed by Holland [30] in 1975. Since then, the genetic algorithm has been popular because it can contribute in finding good solutions for complex mathematical problems, like the VRP and others NP-hard problems.

A genetic algorithm is a randomized search technique operating on a population of individuals (solutions). The search is guided by the fitness value of each individual. The creation of new generation primarily consists of four phases: Representation, Selection, Recombination and Mutation. A simple genetic algorithm can be summarized as follows:

1) *Representation:* Encode the characteristics of each individual in the initial population as chromosome. Set the current population to this initial population.

2) Reproduction: Select two parent chromosomes from the current population. The selection process is stochastic and a chromosome with high fitness is more likely to be selected.

3) *Recombination:* Generate two offspring from the two parents by exchanging pieces of genetic material (crossover).

4) *Mutation:* Apply a random mutation to each offspring with small probability.

5) Repeat steps 2, 3 and 4, until the number of chromosomes in the new population is the same as in the old population.

6) Set the current population to the new population of chromosomes.

This procedure is repeated for a fixed number of generations, or until convergence to a population of similar individuals is obtained. Then, the best chromosome generated during the search is decoded into the corresponding individual. Genetic algorithm, work with a population of candidate solutions instead of just a single solution so they make a multiple way search simultaneously. Each individual represents a potential solution for the problem.

Generally, Genetic algorithm works on maximization, therefore the higher is the objective function the better; the higher the results from a specific individual parameters value, the better will be the fitness of this individual. Additionally, a higher fitness value results in a larger chance for the individual to participate in a crossover, which will generate the individuals in the next generations. In fact, a higher fitness value reflects how intensive was the local search in that region of the search space. Therefore, an individual with good fitness induces a search in that direction. The crossover is done after some kind of individual's selection, partially random based, and partially based on the quality of the individual fitness. The simplest way to do a crossover is breaking up two chromosomes in a random point and exchange them sideways, as illustrated in Fig.1:

0	0	0	1	1	0	0	1	0	1	Parents
0	1	0	1	1	1	0	0	1	1	
0	1	0	1	1	0	0	1	0	1	Result Individuals
0	0	0	1	1	1	0	0	1	1	Next Generation

Fig. 1. Simple crossover

Genetic Algorithms owe their name to an early emphasis on a (binary) encoding and manipulating the genetic makeup of individuals (genotype) rather than using the physical expression of the genetic makeup (phenotype). The practicality of using the genetic algorithm to solve complex optimization problems was demonstrated by De Jong [19] and Goldberg [29].

Thangiah et al [56] were the first to apply genetic algorithm to VRPTW, and uses bit string representation in vehicle routing context. Thangiah [56] described same method in more detail. He describes a cluster-first, route-second method called GIDEON that assigns customers to vehicles by partitioning the customers into sectors by genetic algorithms and customers within each formed sector are routed using cheapest insertion method. In the GIDEON system each chromosome represents a set of possible clustering schemes.

Potvin et al [45] propose a genetic algorithm GENEROUS that directly applies genetic operators to solutions, thus avoiding the coding issues. The fitness values are based on the number of vehicles and total route time. The selection process is stochastic and biased toward the best solution.

Berger [3] proposes a method based on the hybridization of a genetic algorithm with well known construction heuristics. The author omits the coding issues and represents a solution by a set of feasible routes. Bräysy [7], Bräysy [8] proposes several new crossover and mutation operators, testing different forms of genetic algorithms, selection schemes, scaling schemes and the significance of the initial solutions.

Zhu [59] describes a solution to the problem by representing an integer string of length N, where N is the number of customers which are needed to be served. All routes are encoded together, with no special route termination character in between; chromosomes are decoded back into routes based on the customers demand and the vehicle capacity.

Tan et al [54] introduce a genetic algorithm similar to Zhu [59] implementation. The basic strategy of creating the initial population and selection scheme are the same. The well-known Partially mapped crossover (PMX) operator is used to interchange gene material between chromosomes and mutation is performed randomly by swapping nodes. The basic idea in the PMX crossover is to use two crossover points which select two sequences of genes in both parents.

Jung and Moon [34] suggest using the 2D image of a solution for chromosomal cutting within a typical steady-state genetic algorithm. The initial population is created by insertion heuristic of Solomon [51]. Fitness values are based on traveled distance and the selection of parents for mating is performed with the typical binary tournament selection. The 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. A repair algorithm is then 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. The main features of the genetic algorithms proposed for the VRPTW are summarized in Table 1.

2.2. *Evolutionary Strategies*

Evolution strategies or simply ES manipulate population of individuals, which represent 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 genetic algorithm, mutation operators are given a superior role in comparison to the recombination operators. Three important features separate evolutionary strategies from other evolutionary algorithms. First, Evolution strategies use a real-coding of the decision vector, and model the organic evolution at the level of individual's phenotypes. Second, Evolution strategies depend on a deterministic selection of the individuals for the next generation and the search is mainly driven by mutation. Third, the individuals' representation includes a vector of so-called "strategy parameters" in addition to the solution vector and both components are evolved by means of recombination and/or mutation operators.

An individual in Evolution strategies is represented as a pair of real vectors, v = (x,σ). The first vector, x, represents a solution in the search space and consists of real valued variables. The second vector, σ, represents the strategy parameters. In the real-value case, the strategy parameters represent standard deviations of normally distributed random variables. Evolution strategies were developed in the 1970s by Rechenberg [47] and Schwefel [49] to solve optimization problems with real-value variables i.e Evolution strategies were originally developed to solve optimization problems by with real-value decision variables. Especially the (μ, λ)-evolution strategy from Schwefel [49] seems to be a particularly suitable method, since it evolves its so-called strategy parameters according to the Evolution strategies metaphor. Later, in 1990's Hömberger et al [31] introduced new characteristics to solve VRPTW.

Table 1

Author	**Initial Population**	**Fitness**	**Recombination/ Crossover**	**Mutation**
Thangiah[56] [1995]	Random clustering + Insertion heuristics	Routing cost	2-point crossover	Random change of bit values
Potvin et al[45] [1996]	Solomon's insertion	No. of vehicles + Total driving time	Reinsertion of a route or route segment into another parent	Combinations of relocate operator to eliminate routes and Or-opt [42]
Berger et al[3] [1998]	Nearest neighbor	No. of vehicles + Total driving time	Reinsertion with modified Solomon's heuristics	Relocate to reduce the number of routes and nearest neighbor for intra-route improvement
Bräysy[8] [1999]	Random clustering + Solomon's insertion	No. of vehicles + Total driving time + waiting time	Reinsertion with modified Solomon's heuristics	Relocate to reduce the number of routes
Zhu[59] [2000]	Solomon's insertion, λ-interchanges [43], random	Not defined	Random cut-off point and reinsertion into another parent	Reversing the order of a pair or sequence of nodes
Tan et al[54] [2001]	Solomon's insertion, λ-interchanges [43], random	Not defined	Partially mapped crossover (PMX)	Random swap of nodes
Jung et al[34] [2002]	Solomon's insertion + random	Total driving time	Selecting arcs based on 2D image of a solution and nearest neighbor rule selected solution	Random splitting of routes. Or-opt [42], relocation, 2-opt* [44]

Features of Genetic Algorithms for VRPTW

Hömberger et al [31] propose two evolutionary strategies for the VRPTW. The individuals of the starting population are generated by means of a stochastic approach that is based on the savings algorithm of Clark et al [14]. Selection of parents is done randomly and only one offspring is created through the recombination of pair of parents. Thus, a number of $\lambda > \mu$ offspring is created, where μ is the population size. At the end, fitness values are used to select μ offspring for the next generation. 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 search of Or-opt [42], 2-opt* [44] and λ –interchange [43] -move, with λ = 1. In addition, special Or-opt are used for route elimination. The first out of the two proposed metaheursitics evolutionary strategy ES1 skips the recombination phase. The second strategy, ES2, uses the uniform order-based crossover to modify the initially randomly created mutation codes.

Mester [40] proposes that in the beginning, all customers are served by separate routes. Then a set of six initial solutions is created using cheapest reinsertions of single customers with varying insertion criteria. The best initial solution obtained is used as a starting point for the Evolution strategies. 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. The main features of the evolution strategies proposed for the VRPTW are summarized in Table 2.

3. Hybrid Algorithms with Genetic or Evolutionary Components

In this section we review heuristic search methods that hybridize ideas of evolutionary computation with some other search technique, such as tabu search, guided local search and hierarchical tournament selection.

Gehring et al [25] introduced a two phase approach. In the first phase, the evolution strategy ES1 of Homberger et al [31] is applied with a population size of one to minimize the number of routes. In second phase, the total distance is minimized using a

Table 2

Author	**Initial Population**	**Fitness**	**Recombination/ Crossover**	**Mutation**
Homberger et al[31] [1999]	Stochastic saving heuristic	No. of vehicles + Total driving time + elimination of shortest route	Uniform order-based to create sequence for controlling Or-opt	Or-opt, 2-opt* and λ – interchanges, Or-opt for route elimination
Mester[40] [2002]	Cheapest insertion with varying criteria	Not defined	Not defined	Or-opt, 2-opt* and λ – interchanges, GENIUS, modified Large neighborhood search (LNS) of Shaw [50]

Features of Evolutionary Strategies for VRPTW

tabu search algorithm utilizing the same local search operators as ES1. The approach is parallelized using the concept of cooperative autonomy i. e, several autonomous sequential solution procedures cooperate through the exchange of solutions.

Bräysy et al [10] describe two-phase hybrid evolutionary algorithms based on the hybridization of a genetic algorithm and an evolutionary algorithms consisting of several local search and route construction heuristics. In the first phase, a genetic algorithm based on Berger [3] and Bräysy [7] is used to obtain a feasible solution. The algorithm uses a random heuristic to create the initial population, and a Large Neighborhood Search (LNS) based strategy of Shaw [50] within the recombination and mutation phase. The EA used in the second phase picks every combination of two routes in random order and applies randomly one out of four local search operators or route construction heuristic, Or-opt and insertion heuristics.

Wee Kit et al [58] describe a hybrid genetic algorithm, where a simple tabu search based on cross, exchange, relocate and 2-opt neighborhoods, is applied on individual solutions in the later generations to intensify the search. The GA is based on random selection of parent solutions and two 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 common characteristics of parent solutions to offspring by modifying the seed selection procedure and cost function of an insertion heuristic

Bräysy et al [12] continue the work of Bräysy et al [10]. The genetic algorithm of the first phase is replaced with a two-stage multi-start local search of Ibaraki et al [33]. In the first stage, a set of initial solutions is created using a sequential cheapest insertion heuristics. After creating an initial solution, an attempt is made to reduce the number of routes, using a special ejection chain based technique of Glover [28]. In the second phase an evolutionary algorithm is used to minimize the total distance. Here just two crossover operators are used. The first is an extension of CROSS-exchanges of Taillard [53] and other one is similar to the Large Neighborhood Search of Shaw [50], where the search is restricted to two routes only.

Mester et al [41] hybridize the evolution strategies of Mester [40] with 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 ES local search algorithms of Mester [40] in the second stage. The two stages are repeated iteratively until the stopping criterion is met. The composite local search is based on well known relocate, 1-intercange and 2-opt* neighborhoods and the initial solution is created with cheapest insertion heuristic of Mester [40].

Berger et al [4] proposed algorithm relies on the concept of simultaneous evolution of two populations pursuing different objectives subject to partial constraint relaxation. The first population evolves individuals to minimize total distance traveled while second forces on minimizing temporal constraint violation to generate feasible solution, both subject to fixed number of tours. 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.

Alvarenga et al [1] proposes a three phase approach. Initially, a hierarchical tournament selection genetic algorithm is applied. After then, the two phase approach, the genetic and set partitioning, is applied to minimize the travel distance. The *stochastic PFIH* (Push Forward Insertion Heuristic – Solomon [51]) is used to generate the initial population. Nine fitness criteria's 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.

Le Bouthilllier et al [36] propose 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 and four different metaheuristics methods, namely two evolutionary algorithms and two tabu searches. The evolutionary algorithm use 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 [27] and unified tabu search of Cordeau et al [15]. The fitness value of solutions is based on the number of vehicles, distance and waiting times.

Le Bouthillier et al [37] 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. Le Bouthillier et al [37] applied cooperative framework of Le Bouthillier et al [36] to new cooperation parallel method, which is divided into four phases: two phases of diversification at the beginning to broaden the search, and then two intensification phases to focus the search around promising regions. The phases proceed as follows: 1. Use unpromising in-patterns of frequent arcs in the average subpopulation, and prohibit them in the independent metaheuristics. 2. Prohibit arcs from frequent unpromising in-patterns from the worst subpopulation. 3. Work with the average subpopulation, and fix arcs from frequent promising in-patterns. 4. Use frequent promising in-patterns from the elite subpopulation, and fix the arcs for the metaheuristic searches. The first two phases explore pattern lengths in decreasing lengths; the latter two explore them in increasing lengths, by increments of one unit. The main features of the hybrid algorithms with genetic or evolutionary components proposed for the VRPTW are summarized in Table 3.

4. Other Important Hybrid Algorithms

In this section we review other hybrid algorithms which are important to be discussed for comparison with evolutionary algorithms and hybrid algorithms with genetic or evolutionary components.

Ibaraki et al [33] propose local search algorithms for the vehicle routing problem with soft time window constraints. The time window constraint for each customer is treated as a penalty function. 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 at customers so that the total penalty is minimized by using dynamic programming.

Bent et al [2] proposes a two stage hybrid algorithm. 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.

Bräysy [11] present a new deterministic metaheuristics based on a modification of variable neighborhood search of Mladenovic [39]. The proposed procedure is based on new four-phase approach. In this approach a initial solution is first created using new route construction heuristics followed by route elimination procedure to improve solution 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 fourth phase best solution obtained is improved by modifying the objective function to escape from a local minimum.

Chen et al [13] 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.

Csiszár [17] proposes a two phase approach. In the first phase the main focus is on the route elimination. In the second phase focus is on the cost reduction. For this

Table 3

Author	**Phase – I**	**Phase- II**
Gehring et al[25][1999]	ES1 of Homberger et al [31] with μ =1 is used.	Use parallel tabu search utilizing the local search operators of ES1.
Bräysy et al[10] [2000]	Random insertion heuristic, modification of LNS, reinsertion with modified Solomon's heuristic.	Modified LNS, modified CROSS exchanges, Or-opt, insertion heuristic, relocate.
Wee Kit et al[58] [2001]	Random selection, reordering by first crossover operator, modification of Solomon's insertion heuristic by second crossover operator.	Use tabu search based on CROSS exchange, relocate and 2-opt* neighborhoods.
Bräysy et al[12] [2003]	A two-stage multi-start local search of Ibaraki et al [33], cheapest insertion heuristics, reordering the routes in the ejection chain of Glover [28].	Modified LNS of Shaw [50], modified CROSS exchanges of Taillard [53].
Mester et al[41] [2005]	Cheapest insertion heuristics, guided local search, relocate, 1-interchange, 2-opt* neighborhood	Modified evolution strategies local search algorithm of Mester [40]. Filling procedure of Bent et al [2] is used.

Features of Hybrid Algorithms with Genetic or Evolutionary Components for VRPTW

a new model called "Magic Bricks" is proposed. The model suggests let the width of a brick is 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 primarily smaller wall - with fewer routes – secondly try to reduce the length of the brick walls.

Ropke et al [48] presented a new approach of Adaptive Large Neighborhood Search (ALNS), an extension of LNS by Shaw [50] with an adaptive layer. This layer adaptively chooses among a number of insertion and removal heuristics, to intensify and diversify the search. ALNS is a local search framework in which a number of simple algorithms compete to modify the current solution. 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 (which can be simulated annealing or tabu search or guided local search) applied at master level. It first transforms a VRPTW instance to a rich pickup and delivery problem with time windows (RPDPTW) and then solved it using ALNS.

Sontrop et al [52] introduces 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. The main features of the other important hybrid algorithms proposed for the VRPTW are summarized in Table 4.

5. Analysis of Results

In this section we compared and analyzed the above described metaheuristics algorithms, using the results obtained for Solomon's [51] well known 56 benchmark problems and to the extended benchmark problems developed by Gehring et al [25].

5.1. *Solomon's Problem Instances*

Solomon's [34] test problems have been found the most common way to assess and compare the value of the various heuristics approaches proposed in the literature. These problems have a hundred customers, a central depot, capacity constraints, time windows on the time of delivery, and a total route time constraint. The C1 and C2 classes have customers located in clusters and in the R1 and R2 classes the customers are at random positions. The RC1 and RC2 classes contain a mix of both random and clustered customers. Each

Table 4

Author	Phase – I	Phase- II
Bent et al[2] [2004]	Simulated annealing to minimize the number of routes.	LNS of Shaw [50] to minimize total travel cost.

Features of Other Important Hybrid Algorithms for VRPTW

class contains between 8 to 12 individual problem instances and all problems in any class have the same customer locations and the same vehicle capacities, only time window differs. In the term of time window density, the problems have 25%, 50%, 75% or 100% of customers with time windows. C1, R1, RC1 problems have a short scheduling horizon, and require 9 to 19 vehicles. Short horizon problems have vehicles that have small capacities and short route times, and cannot service many customers at one time. Classes C2, R2 and RC2 are more representative of "long-haul" delivery with longer scheduling horizons and fewer 2-4 vehicles. Both travel time and distance are given by the Euclidean distance between points. The CNV/CTD indicates the cumulative number of vehicles (CNV) and cumulative total distance (CTD). In Table 5, we list computational results for the methods for which computing times have been reported in a systematic fashion.

5.2. *Gehring and Homberger Problem Instances*

The Gehring et al [25] extended benchmark problems were constructed similar to the 100 customer problem instances by Solomon [51]. They consist of 5 sets of 200, 400, 600, 800 and 1000 customers, with 60 instances in each set.

In Table 6, we present results for the 300 extended problems developed by Gehring et al [25]. Table 6 shows cumulative number of vehicles (CNV) and cumulative total distance (CTD), for each problem size of 200, 400, 600, 800 and 1000 customers. Only a few authors have tackled these extended benchmarks. So, we are presenting all those methods, we are aware of reporting results to solve Gehring et al [25] extended benchmark problems. These methods are mostly hybrid algorithms, Homberger et al [32] and Gehring et al [25,26] shows hybridization of ES with tabu search, Bräysy et al [12] presents a hybrid of multi start local search with ejection chain approach, Li et al [38] shows hybridization of local search with simulated annealing, Bent et al [2] hybrids simulated annealing and LNS, Mester et al [41] shows hybridization of ES with guided local search, Le Bouthillier et al [36] presents hybrid of two evolutionary algorithms and two tabu search, Le Bouthillier et al [37] extended the previous work with pattern identification and Ropke et al [48] hybrids ALNS with either simulated annealing or guided local search. All methods consider a hierarchical objective. According to this objective, Homberger et al [32] shows best performance for CNV for 200 and 800 customer problems, Mester et al [34] performs best results for CTD for 200,400,600,800 and 1000 customer problems and also for CNV for 200 customers, while Ropke et al [48] shows the best performance for CNV for 200, 400, 600 and 1000 customer problems.

6. Conclusion

The VRP and VRPTW, belonging to the class of the NP-hard combinatorial optimization problems, require heuristic solution strategies for most real life instances. In this paper we have surveyed the evolutionary algorithms for VRPTW methodologies and then compare them with other metaheuristics such as tabu search and ant colony systems. For the Solomon's problem instance evolutionary strategies of Mester [40], hybridization of evolutionary algorithms with other search techniques by Homberger et al [32], and Le Bouthillier et al [37] seems to achieve the best performance. In other important hybrid algorithms Bent et al [2] and Ropke et al [48] shows best performance. Homberger et al [32] performs best for R1, Bent et al [2] performs best for RC, CNV and CTD, while Mester [40] performs best in R2 and RC2. For C1 and C2, almost all papers report optimal solution. Evolutionary algorithms and their hybrids show best performance as compared to tabu search, ant colony system and other important hybrid algorithms. To summarize, it seems that it is important to include special methods for route reduction, combine several different search methods, employ some improvement heuristics, and create a high quality initial population to achieve the best robustness.

Table 5

Authors	**R1**	**R2**	**C1**	**C2**	**RC1**	**RC2**	**CNV/ CTD**	**TIME**
Ant Algorithms								
Gambardella et al[24][1999]	12.00 1217.73	**2.73** 967.57	**10.00** **828.38**	**3.00** **589.86**	11.63 1382.42	**3.25** 1129.19	407 57525	Sun U1,167 MHz,-,-
Tabu Search								
Rochat et al[36] [1995]	12.25 1208.50	2.91 961.72	**10.00** **828.38**	**3.00** **589.86**	11.88 1389.22	3.38 1117.44	415 57231	SG 100 MHz, 1run, 92.2 min
Taillard et al[53] [1997]	12.17 1209.35	2.82 1016.58	**10.00** **828.38**	**3.00** **589.86**	**11.50** 1389.22	3.38 1117.44	410 57523	Sun S10,-,-,-
Brando et al[6] [1999]	12.58 1205	3.18 995	**10.00** 829	3.00 591	12.13 1371	3.50 1250	425 58562	----------
Cordeau[15] [2001]	12.08 1210.14	2.73 969.57	**10.00** **828.38**	**3.00** **589.86**	**11.50** 1389.78	**3.25** 1134.52	407 57556	Sun U2 300 MHz,-,-
Genetic Algorithms								
Thangiah[56] [1995]	12.75 1300.25	3.18 1124.28	**10.00** 892.11	3.00 749.13	12.50 1474.13	3.38 1411.13	429 65074	Solbourne 5/802, -, 2.1 min.
Potvin et al[35] [1996]	12.58 1296.83	3.00 1117.64	**10.00** 838.11	3.00 590.00	12.13 1446.25	3.38 1368.13	422 62634	Sun S10,-, 25 min.
Berger et al[3] [1998]	12.58 1261.58	3.09 1030.01	**10.00** 834.61	3.00 594.25	12.13 1441.35	3.50 1284.25	424 60539	Sun S10,-, 1-10 min.
Bräysy[8] [1999]	12.58 1272.34	3.09 1053.65	**10.00** 857.64	3.00 624.31	12.13 1417.05	3.38 1256.80	423 60962	Sun U Enterprise, 450, 5runs, 17 min.
Tan et al[54] [2001]	13.17 1227	5.00 980	10.11 861	3.25 619	13.50 1427	5.00 1123	478 58605	PII 330 MHz, -, 25 min.
Jung et al[34] [2002]	13.25 1179.95	5.36 878.41	**10.00** **828.38**	**3.00** **589.86**	13.00 1343.64	6.25 1004.21	486 54779	PIII 1 GHz, 100 runs, 0.8 min.
Evolutionary Strategies								
Homberger et al[31] [1999]	**11.92** 1228.06	**2.73** 969.95	**10.00** **828.38**	**3.00** **589.86**	11.63 1392.57	**3.25** 1144.43	406 57876	P200 MHz, 10 runs, 13 min.
Mester[30] [2002]	12.00 1208	**2.73** **954**	**10.00** 829	3.00 590	**11.50** 1387	**3.25** **1119**	406 57219	PIII 450MHz, -, 150.2 min.
Hybrid Algorithms with Genetic or Evolutionary Components								
Gehring et al[25] [1999]	12.42 1198	2.82 947	**10.00** 829	3.00 590	11.88 1356	**3.25** 1140	415 56942	4*P200 MHz, 1 run, 5 min.
Bräysy[10] [2000]	12.42 1213.86	3.09 978.00	**10.00** 828.75	3.00 591.81	12.13 1372.20	3.38 1170.23	421 57857	Celeron 366 MHz, 5 runs, 15 min.
Gehring et al[26] [2001]	12.00 1217.57	**2.73** 961.29	**10.00** 828.63	3.00 590.33	**11.50** 1395.13	**3.25** 1139.37	406 57641	4*P400 MHz, 5 runs, 13.5 min.
Homberger et al[32] [2001]	**11.92** **1212.73**	2.73 955.03	**10.00** **828.38**	**3.00** **589.86**	**11.50** 1386.44	**3.25** 1123.17	**305** 57309	P400 MHz,-,-

contd...

Table 5

Authors	**R1**	**R2**	**C1**	**C2**	**RC1**	**RC2**	**CNV/ CTD**	**TIME**
Wee Kit et al[58] [2001]	12.58 1203.32	3.18 951.17	**10.00** 833.32	3.00 593.00	12.75 1382.06	3.75 1132.79	432 57265	DW 433a,-, 147.4 min.
Bräysy et al[12] [2003]	12.00 1220.14	**2.73** 977.57	**10.00** **828.38**	3.00 589.86	**11.50** 1397.44	**3.25** 1140.06	406 57870	AMD 700MHz, 3 runs, 9.1 min.
Berger et al[3] [2003]	**11.92** 1221.06	**2.73** 975.43	**10.00** 828.48	3.00 589.93	**11.50** 1389.89	**3.25** 1159.37	**305** 57952	P400 MHz, -, 30 min.
Alvarenga[1] [2005]	**11.92** 1224	**2.73** 1012	**10.00** 828.4	3.00 590.9	**11.50** 1417	**3.25** 1195	**305** 58912	n/a
Le Bouthillier et[36] al [2005]	12.08 1209.19	**2.73** 963.62	**10.00** **828.38**	**3.00** **589.86**	**11.50** 1389.22	**3.25** 1143.70	407 57412	5*P850 MHz, 1 run, 12 min.
Le Bouthillier et[37] al [2005]	**11.92** 1214.20	**2.73** 954.32	**10.00** **828.38**	**3.00** **589.86**	**11.50** 1385.30	**3.25** 1129.43	**305** 57360	5*P850 MHz, 1 run, 12 min.
Other Important Hybrid Algorithms								
Bräysy[11] [2003]	**11.92** 1222.12	**2.73** 975.12	**10.00** **828.38**	**3.00** **589.86**	**11.50** 1389.58	**3.25** 1128.38	**305** 57710	P200 MHz, 1 run, 82.5 min.
Ibaraki et al[33] [2003]	**11.92** 1217.40	**2.73** 959.11	**10.00** **828.38**	**3.00** **589.86**	**11.50** 1391.03	**3.25** 1122.79	**305** 57444	PIII 3 GHz,-,-
Bent et al[2] [2004]	12.18 1231.08	**2.73** 954.18	**10.00** **828.38**	**3.00** **589.86**	**11.50** **1384.17**	**3.25** 1124.46	**305** **57272**	Sun U10, 440MHz, 5 runs, 120 min.
Chen et al[13] [2005]	12.83 1203.56	3.09 932.23	**10.00** 828.76	**3.00** **589.86**	12.50 1363.84	3.75 1079.81	432 56429	PIII 1000 MHz,-,-
Ropke et al[38] [2005]	**11.92** 1213.39	**2.73** 958.60	**10.00** **828.38**	**3.00** **589.86**	**11.50** 1385.39	**3.25** 1124.77	**305** 57360	PIV 3 GHz,-,-

Comparison of Evolutionary Algorithms, Metaheuristics, Hybrid Algorithms with genetic or evolutionary components and other important Hybrid Algorithms for Solomon's Problem Instances. The best results are in boldface.

Table 6

Authors		**200**	**400**	**600**	**800**	**1000**
Evolutionary Strategies						
Homberger et al[31] [1999]	CNV CTD	**694** 173,313	1388 409,764	2076 851,681	**2755** 1,479,802	3461 2,236,583
Hybrid Algorithms with Genetic or Evolutionary Components						
Gehring et al[25] [1999]	CNV CTD	**694** 176,180	1390 412,270	2082 867,010	2770 1,515,120	3461 2,276,390
Gehring et al[26] [2001]	CNV CTD	696 179,328	1392 428,489	2079 890,121	2760 1,535,849	3446 2,290,367
Homberger et al[32] [2001]	CNV CTD	699 180,602	1397 431,089	2088 890,293	2773 1,516,648	3459 2,288,819
Bräysy et al[12] [2003]	CNV CTD	695 172,406	1391 399,132	2084 810,662	2776 1,384,306	3465 2,133,376
Le Bouthillier et al[36] [2005]	CNV CTD	**694** 173,062	1390 410,330	2088 840,583	2766 1,475,436	3451 2,225,367
Mester et al[41] [2005]	CNV CTD	**694** **168,573**	1389 **390,386**	2082 **796,172**	2765 **1,361,586**	3446 **2,078,110**
Le Bouthillier et al[37] [2005]	CNV CTD	**694** 169,959	1389 396,612	2086 809,494	2761 1,443,400	3442 2,133,645
Other Important Hybrid Algorithms						
Li et al[38] [2003]	CNV CTD	707 172,472	1414 405,656	2112 843,320	2802 1,416,531	3490 2,176,398
Bent et al[2] [2004]	CNV CTD	697 171,715	1393 410,112	2091 858,040	2778 1,469,790	3468 2,266,959
Ropke et al[48] [2005]	CNV CTD	**694** 169370	**1385** 395,970	**2071** 818,863	2758 1,372,619	**3438** 2,146,752

Comparison of Evolutionary Algorithms, Hybrid Algorithms with genetic or evolutionary components and other important Hybrid Algorithms for the extended problem instances of Gehring and Homberger. The best results are in boldface.

References

[1] Alvarenga, G.B., Silva, R.M.A., Sampaio, R.M.,"A Hybrid Algorithm for the Vehicle Routing Problem with Time Window", In *Proceedings of Fifth internal Conference on Hybrid Intelligent Systems*, Rio De Janeiro, Brazil, 2005.

[2] Bent, R., Hentenryck, P., "A Two-stage Hybrid local search for the Vehicle Routing Problem with Time Windows", *Transportation Sciences* 38(4) : 515-530, 2004.

[3] Berger, J., Salois, M., Begin, R., "A Hybrid Genetic Algorithm for the Vehicle Routing Problem with Time Windows", In *Proceedings of the 12th Bienneal Conference of the Canadian Society for Computational Studies of Intelligence, 114-127, Springer-Verlag,* Berlin 1998.

[4] Berger, J., Barkaoui, M., Bräysy, O., "A Route-directed Hybrid Genetic Approach for the Vehicle Routing Problem with Time Windows", *Information Systems and Operation Research* 41, 179-194, 2003.

[5] Box, G.E.P., "Evolutionary Operation: A Method of Increasing Industrial Productivity", *Applied Statistics* 6: 81 -101, 1957.

[6] Brando, J.," Meta Heuristic for the Vehicle Routing Problem with Time Windows". In *Meta Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, 19-36, Kluwer, Boston, MA, 1999.

[7] Bräysy, O., "A Hybrid Genetic Algorithm for the Vehicle Routing Problem with Time Windows", Unpublished, Vaasa University, Vaasa, Finland, 1999.

[8] Bräysy, O., "A New Algorithm for the Vehicle Routing Problem with Time Windows based on Hybridization of a Genetic Algorithm and Route Construction Heuristics", *Proceedings of the University of Vaasa*, Research Papers 227, Vaasa, Finland, 1999.

[9] Bräysy, O., "Genetic Algorithms for the Vehicle Routing Problem with Time Windows", *Arpakannus1/2001, Special issue on Bioinformatics and Genetic Algorithms*: 33-38, 2001.

[10] Bräysy, O., Berger, J., Barkaoui, M., "A New Hybrid Evolutionary Algorithm for the Vehicle Routing Problem with Time Windows", *Presented at the Route 2000-workshop*, Skodsborg, Denmark, August 2000.

[11] Bräysy, O., "A Reactive Variable Neighborhood Search for the Vehicle Routing Problem with Time Windows", *INFORMS Journal on Computing,* 15:347-368, 2003.

[12] Bräysy, O., Dullaert, W., "A Fast Evolutionary Metaheuristic for the Vehicle Routing Problem with Time Windows", *International Journal on Artificial Intelligence Tools* 12: 153-172, 2003.

[13] Chen, C. H., Ting, C.J., "A Hybrid Ant Colony System for Vehicle Routing Problem with Time Windows", *Journal of the Eastern Asia Society for Transportation Studies* 6: 2822-2836, 2005.

[14] Clarke, G., Wright, J. V., "Scheduling of Vehicles from the Central Depot to a number of delivery points", *Operations Research* 12:568-581, 1964.

[15] Cordeau, J.F., Laporte, G., Mercier, A., "A Unified Tabu Search for Vehicle Routing Problem with Time Windows", *Journal of the Operational Research Society* 52: 928-936, 2001.

[16] Cordeau, J.F., Laporte, G., Mercier, A., "Improved Tabu Search Algorithm for the handling of route duration constraints in Vehicle Routing Problems with Time Windows", *Journal of the Operational Research Society* 55(5): 542-546, 2004.

[17] Csiszár, S., "Route Elimination Heuristics for Vehicle Routing Problems with Time Windows", *Acta Polytechnica Hungarica* 2(2): 77-89, 2005.

[18] Darwin, C., "On the Origin of Species", *1st edition, Harvard University Press*, MA. 1859.

[19] De Jong, K.A., "An Analysis of the Behavior of a Class of Genetic Adaptive Systems", *Ph.D. thesis*, University of Michigan, U.S.A, 1975.

[20] Duhamel, C., Potvin, J.Y., Rousseau, J.M. , "A Tabu Search Heuristic for the Vehicle Routing Problem with Backhauls and Time Windows", *Transportation Science* 31: 49-59, 1999.

[21] Fisher, M. L.," Optimal Solution of Vehicle Routing Problems Using Minimum K-trees", *Operations Research* 42:626-642, 1994.

[22] Forgel, L.J., Owens, A., Walsh, M.J., "Artificial Intelligence through Simulated Evolution". New York: Wiley Publishing. 1966.

[23] Fraser, A.S., "Simulation of Genetic System by Automatic Digital Computers", *Australian Journal of Biological Science* 10: 484-491, 1957.

[24] Gambardella, L. M., Taillard, E., Agazzi, G.," MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows", *New Ideas in Optimization*, McGraw-Hill, 1999.

[25] Gehring, H., Hömberger, J., "A Parallel Hybrid Evolutionary Metaheuristics for Vehicle Routing Problem with Time Windows", In *Proceedings of EUROGEN99*. Jyväskylä: University of Jyväskylä, 57-64, 1999.

[26] Gehring, H., Hömberger, J., "Parallelization of a Two-Phase Metaheuritic for Routing Problem with Time Windows", *Asia-Pacific Journal of Operation Research* 18, 35-47, 2001.

[27] Gendreau, M., Hertz, A., Laporte, G., "A Tabu Search Heuristics for the Vehicle Routing Problem", *Management Science* 40, 1276-1290, 1994.

[28] Glover, F., "New Ejection Chain and Alternating Path Methods for Traveling Salesman Problems", *Computer Science and Operations Research: New Developments in Their Interfaces*. Oxford: Pergamon Press, 449-509, 1992.

[29] Goldberg, D., Korb, B., Deb, K., "Messy Genetic Algorithms: Motivation, Analysis and First Results", *Complex Systems* 3: 493-530, 1989.

[30] Holland, J.H., "Adaptation in Natural and Artificial System". Ann Arbor, Michigan: The University of Michigan Press, 1975.

[31] Hömberger, J., Gehring, H., "Two Evolutionary Meta Heuristics for the Vehicle Routing Problem with Time Windows". *INFORMS Journal on Computing*, 37(3):297-318, 1999.

[32] Hömberger, J., Gehring, H., "A Two-Phase Hybrid Metaheuristic for the Vehicle Routing Problem with Time Windows", Working Paper, Department of Economics, University of Hagen, Germany, 2001.

[33] Ibaraki, T., Imahori, S., Kubo, M., Uno, T., Yagiura, M., "Effective Local Search Algorithms for Routing and Scheduling Problems with General Time Window Constraints", Forthcoming in *Transportation Science*, 2003.

[34] Jung, S., Moon, B. R., " A Hybrid Genetic Algorithm for the Vehicle Routing Problem with Time Windows", In *Proceddings of Genetic and Evolutionary Computation Conference*, San Franciso, Morgan Kaufmann Publishers : 1309-1316, 2002

[35] Kohl, N., Desrosiers, J., Madsen, O.B., Solomon, M.M., Soumis, F., "2-Path Cuts for the Vehicle Routing Problem with Time Windows," *Transportation Science* 33:101-116, 1999.

[36] Le Bouthillier, A., Crainic, T.G, "Co-operative Parallel Metaheuristice for the Vehicle Routing Problem with Time Window", *Computers & Operation Research* 32(7): 1685-1708, 2005.

[37] Le Bouthillier, A., Crainic, T.G, Kropf, P., "A Guided Cooperative Search for the Vehicle Routing Problem with Time Windows", *IEEE Intelligent systems* 20(4): 36-42, July/August 2005.

[38] Li, H., Lim, A., "Local Search with Annealing-Like Restarts to Solve the VRPTW" *European Journal of Operational Research* 150: 115-127, 2003.

[39] Mladenovic, N., Hensen P., "Variable Neighborhood Search", *Computers & Operations Research* 24: 1097-1100, 1997.

[40] Mester, D. ,"An Evolutionary Strategies Algorithm for Large Scale Vehicle Routing Problem with Capacity and Time Window Restrictions", Working paper, Institute of Evolution, University of Haifa, Israel, 2002.

[41] Mester, D., Bräysy, O., "Active Guided Evolution Strategies for Large Scale Vehicle Routing Problem with Time Window", *Computers & Operation Research* 32(6):1593-1614, 2005.

[42] Or, I., "Traveling Salesman-type Combinatorial Optimization Problems and their relation to the logistics of regional blood banking", Ph.D. dissertation, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, 1976.

[43] Osman, I. H., "Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem", *Annals of Operation Research*, 41:421-451, 1993

[44] Potvin, J., Rousseau, J.M.,"An Exchange Heuristic for Routing Problems with Time Window", *Journal of Operational Research Society*, 46:1433-1446, 1995.

[45] Potvin, J., Bengio. S. "The Vehicle Routing Problem with Time Windows part II: Genetic Search. *INFORMS Journal on Computing*, 8(2):165-172, 1996.

[46] Rochat, Y., Taillard, E., "Probabilistic Diversification and Intensification in Local Search for Vehicle Routing." *Journal of Heuristics* 1: 147-167, 1995.

[47] Rochenberg, I., "Evolutions Strategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution", Frommann-Holzboog, Stuttgart, 1973.

[48] Ropke, S., Pisinger, D., "A General Heuristics for Vehicle Routing Problems", To appear in *Computers & Operations Research*, 2005.

[49] Schwefel, H.P., "Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie" Birkhäuser, Basel, 1977.

[50] Shaw, P., "Using Constraint Programming and Local Search Methods to solve Vehicle Routing Problems", *Principles and Practice of Constraint Programming*, Springer-Verlag , 417-431, 1998.

[51] Solomon, M.M., "Algorithms for Vehicle Routing and Scheduling Problems with Time Window Constraints", *Operations Research* 35:254-265, 1987.

[52] Sontrop, H.M.J., Horn, S.P. van der, Teeuwen, G., Uetz, M., "Fast Ejection Chain Algorithms for Vehicle Routing with Time Windows", In *Proceedings of Hybrid Metaheuristics: Second International Workshop"*, Springer-Verlag 3636, 2005.

[53] Taillard, E.D., Badeau, P., Gendreau, M., Guertin, F., Potvin, J., "A Tabu Search Heuristics for the Vehicle Routing Problem with Soft Time Windows". *Transportation Science* 31:170-186, 1997.

[54] Tan, K. C., Lee, L. H., Ou, K., "Hybrid Genetic Algorithms in Solving Vehicle Routing Problems with Time Window Constraint", *Asia-Pacific Journal of Operation Research* 18:121-130, 2001.

[55] Thangiah, S., Osman, I. H.,Sun, T.," Hybrid Genetic Algorithm, Simulated Annealing and Tabu Search Methods for Vehicle Routing Problems with Time Windows", *Technical Report UKC/OR94/4,* Institute of Mathematics and Statistics, University of Kent, Canterbury, UK, 1994.

[56] Thangiah, S., "Vehicle Routing with Time Windows using Genetic Algorithms", In *Application Handbook of Genetic Algorithms: New Frontiers*, volume II, pp: 253-277, CRC Press, Boca Raton, 1995.

[57] Toth, P., Vigo, D., "The Vehicle Routing Problem, Monographs of Discrete Mathematics and Applications", $S.I.A.M$, Philadelphia, PA, 2002.

[58] Wee Kit, H., Chin, J., Lim, A., "A hybrid Search Algorithm for the Vehicle Routing Problem with Time Windows", *International Journal on Artificial Intelligence Tools* 10, 431-449, 2001.

[59] Zhu, K., "A New Genetic algorithm for VRPTW", *International Conference on Artificial Intelligence*, Las Vegas, USA

Received 5 January 2005; revised 25 March 2006; accepted 20 April 2006