Chapter 4

Heuristics, metaheuristics, and hyperheuristics for rich vehicle routing problems

Miroslaw Blocho
ABB, Krakow, Poland

4.1 Heuristics for rich vehicle routing problems

The heuristics is an approach to solve a given problem that does not guarantee obtaining the optimal solution. However, they allow us to elaborate high-quality feasible solutions that meet the problem objectives. The main families of heuristics are classical heuristics developed mostly between 1960 and 1990 and metaheuristics with rapid development lasting until now [103].

The classical heuristics can be grouped into two main groups, construction and improvement techniques. Both methods perform a limited exploration of the solution space and usually produce good quality solutions within reasonable time. Most of them can be relatively easily extended for various constraints encountered in real life, and for this reason, they are still used in many commercial applications [103].

The main reason to develop and use heuristic approaches is finding good-quality solutions of various problems in short time. Other reasons are the nonexistence of an exact method to solve the problem and flexibility of a heuristic algorithm to handle additional side constraints. Summing up, a heuristics is considered "good" when a solution can be computed with reasonable computational effort, the resulting solution is near-optimal, and the low probability of obtaining solution that is far from optimal [111].

4.1.1 Construction heuristics

Construction heuristics build feasible solutions while trying to minimize its cost but often do not consider any improvement phases. Route construction heuristics usually start from an empty solution and build iteratively subsequent routes by inserting customers one by one until all customers are served. They are considered the simplest and also fastest methods to solve the problem; however, resulting solutions are usually far from optimal solutions. Due to that fact, they

Smart Delivery Systems. https://doi.org/10.1016/B978-0-12-815715-2.00009-9

Copyright © 2020 Elsevier Inc. All rights reserved.

are commonly used only as the baseline to find an initial solution to the problem, which is later improved in the next phases of the algorithm [103]. The main methods used in construction heuristics for the routing problems are nearest-neighbor, insertion, and savings techniques.

In the nearest-neighbor method, the solution is constructed by a greedy algorithm of selecting the closest unserved customer each time. This method was proposed in 1956 by Flood [52] and is considered the simplest way of creating initial solutions. Although the method was proposed many years ago, it still attracts interest. Kizilateş and Nuriyeva [94] provided in 2013 a modification of the nearest-neighbor algorithm for the traveling salesman problem. In 2015, Joshi and Kaur proposed the nearest-neighbor insertion algorithm for solving Capacitated Vehicle Routing Problem (CVRP), where they investigated a practical scenario in which different college buses take students from different bus stops.

Insertion heuristics generate feasible solutions by repeatedly inserting unserved customers into partial feasible routes. Various variants of the insertion heuristics were created due to two key decisions made at every iteration: selection of customer to be inserted and choosing a place of insertion [31]. Insertion heuristics proved to be very fast and able to produce relatively good initial solutions. On the other hand, they are usually easy to be implemented and can be easily extended to handle more complex constraints. Most of the insertion methods were created several decades ago for the family of the vehicle routing problems by Solomon [172], Vigo [192], Liu and Shen [106], and Salhi and Nagy [164]. In 2006, Joubert and Classen [87] proposed a new sequential insertion heuristic for the initial solution to a constrained VRP introducing a new concept of time window compatibility. More recently, in 2015, Pinto et al. [147] proposed an insertion heuristics for the CVRP with loading constraints and mixed linehauls and backhauls. Their approach was based on standard insertion heuristics extended to tackle the explicit consideration of loading constraints.

The savings algorithm was originally developed for the VRP with one central repository and variable number of vehicles by Clarke and Wright [35] in 1964. Its main idea is to compute savings for combining two customers into the same route. The main advantage of that algorithm is that it can be applied for both directed and undirected problems. The savings method was proposed more than fifty years ago, but it still attracts some interest. Gajpal and Abad [56] proposed in 2010 saving-based algorithms for the vehicle routing problem with simultaneous pickup and delivery introducing cumulative net-pickup approach for checking the feasibility when two existing routes are merged. In 2012, Pichpibul and Kawtummachai [146] proposed a new enhancement for Clarke–Wright savings algorithm to optimize the CVRP. They suggested two-phase probabilistic mechanism and the route postimprovement, which gave good results for benchmarks involving from 16 to 135 customers.

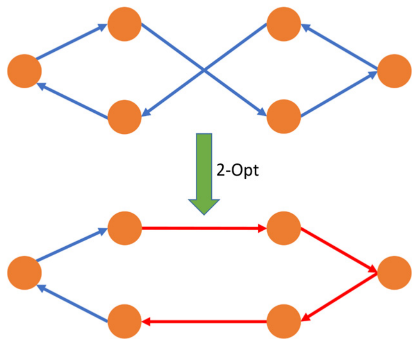

FIGURE 4.1 2-Opt local search move.

4.1.2 Improvement heuristics

The improvement heuristics are usually used for already generated solutions by other heuristics or exact algorithms. Local search methods are typically applied for simple local modifications such as customer or arc exchanges to generate neighboring solutions of possibly better quality. In case better solution is found, it replaces the current one, and the process is continued until local minimum has been found [103]. The main idea driving local search is that by repeatedly improving the quality of solution by making small changes (called moves) it is often possible to find very good solutions. Different heuristics are mostly characterized by different types of local search moves and by obtained neighborhoods. Considering created neighborhoods, there are usually two strategies considered, first-accept (FA) and best-accept (BA). In the FA strategy the first neighbor satisfying acceptance criteria is selected, whereas in the BA strategy, all neighbors have to be examined, and the best one is chosen [27].

The most famous local search moves are 2-Opt and 3-Opt related to edge-exchange neighborhoods for a single route. The 2-Opt method was proposed by Flood [52] in 1956 and Croes [39] in 1958 and is based on deleting two edges from the same route followed by reconnecting the subtours in the other way. A sample 2-Opt local search move is shown in Fig. 4.1. The 3-Opt neighborhood was proposed by Bock [20] in 1958 and is similar to 2-Opt except that three edges are deleted and rejoined with three new links. A sample 3-Opt local search move is shown in Fig. 4.2. In turn, k-Opt neighborhood is a generalization of the 2-Opt and 3-Opt neighborhoods to k-edge exchanges in one local search move. It is important to note that checking k-Opt neighborhoods requires $O(n^k)$ computational time. The first works for application of k-Opt improvement heuristics for the vehicle routing problems were conducted by Russel [161] in 1977. They were further improved mostly in the area of checking the infeasible neighboring solutions by Savelsbergh [165], Solomon and Desrosiers [173], and Baker and Schaffer [10].

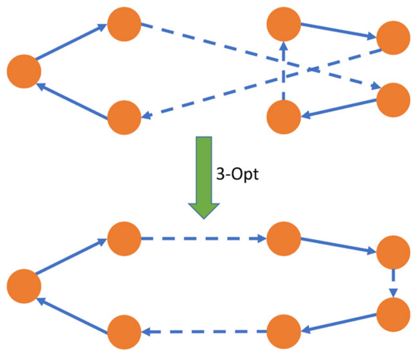

FIGURE 4.2 3-Opt local search move.

The OR-Opt technique proposed in 1976 by Or [134] is a modification of the 3-Opt by taking into account only such 3-Opt exchanges that result in one, two, or three adjacent customers inserted between two other customers. Sample OR-Opt local search move is shown in Fig. 4.4. Replacing up to three edges in the original tour by three new edges does not modify the orientation of the route, which is very crucial in many vehicle routing problems. The cross-exchange neighborhood is constructed by exchanging two subroutes of length at most k, and thus it may be considered as a specific subset of the k-Opt neighborhood [178].

In 1992, Savelsbergh [165] proposed interroute operators for relocation and exchange. The relocate operator is used to move one customer from one route to another, whereas the exchange operator swaps two customers in two different routes. The 2-Opt* neighborhood is a specific variant of 2-Opt, which is a set of solutions obtained by removing two edges from two different routes, followed by adding two different edges to reconnect broken subroutes [153]. A sample 2-Opt* local search move is shown in Fig. 4.5.

This type of neighborhood was introduced in 1995 by Potvin and Rousseau [153], who were doing extensive studies on 2-Opt, 3-Opt, and OR-Opt techniques. The key point in 2-Opt* moves is that the order of customers in subroutes is maintained. Potvin and Rousseau proposed also a hybrid approach comprising both OR-Opt and 2-Opt* methods together, which proved to be competitive to these methods applied separately. In this hybrid approach the operator (OR-Opt, 2-Opt*) changes each time local minimum is reached.

The more specialized neighborhoods are GENI-exchange by Gendreau et al. [62], λ-interchange by Osman [140], CROSS-exchange by Taillard et al. [178], and ejection chains by Glover [66]. The GENI operator is an extension of the standard relocate operator by Savelsbergh with such a difference that an out-relocated customer may be inserted between two customers in the destination route even violating the order of the customers in that route. A sample GENI exchange is shown in Fig. 4.3. The λ-interchange defines the neighbor-

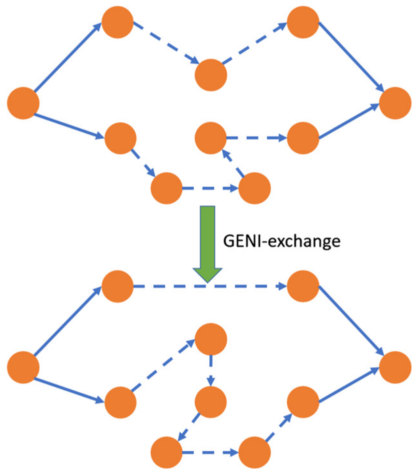

FIGURE 4.3 GENI local search move.

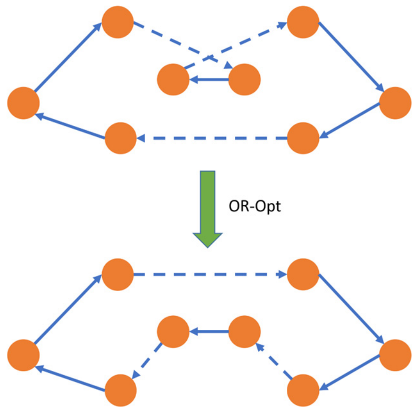

FIGURE 4.4 OR-Opt local search move.

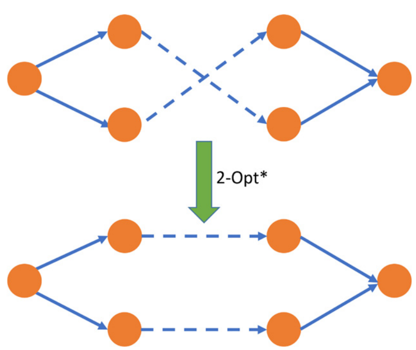

FIGURE 4.5 2-Opt* local search move.

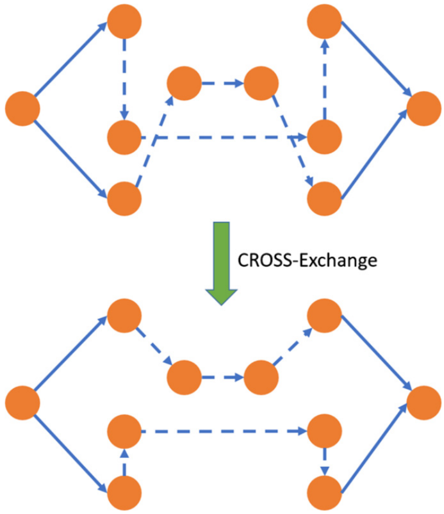

FIGURE 4.6 CROSS-exchange local search move.

hood with λ customers shifted from one route to another or exchanged between two routes with a specific order on each pair of routes. The basic idea of the CROSS-exchange is to exchange two consecutive customers from one route with two consecutive customers from the second route preserving the original order. A sample CROSS-exchange is shown in Fig. 4.6. In turn, ejection chains are based on a sequence of removal and insertion moves repeated until an unrouted customer can be inserted into the destination route without the need to eject any customer.

4.2 Metaheuristics for rich vehicle routing problems

Metaheuristics is an enhancement of classical heuristics with emphasis on deep exploration of the solution space. They usually combine sophisticated neighborhood search rules and recombination of solutions [103]. The quality of the solutions produced by metaheuristics is typically much higher than that obtained by classical heuristics techniques but with a price of increased computing time. The metaheuristic techniques are context dependent and usually require finely tuned parameters, which unfortunately make their extensions to other problems difficult. Many various metaheuristics have been proposed for the vehicle routing problems, and they can be widely divided into local search, population search, and learning mechanism groups; however, best metaheuristics merge ideas from different approaches [38]. These different approaches will be described in the following subsections.

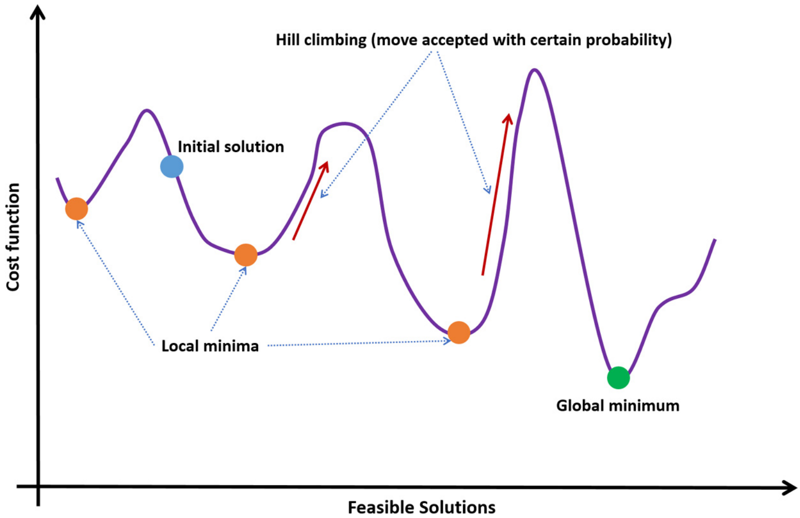

FIGURE 4.7 Simulated Annealing procedure (blue (mid gray in print version) dot – initial solution, orange (light gray in print version) dot – local minimum, green (mid gray in print version) dot – global minimum, red (dark gray in print version) arrow – hill climbing (move accepted with certain probability)).

4.2.1 Simulated Annealing

Simulated Annealing (SA) is a probabilistic technique proposed in 1983 by Kirkpatrick et al. [93] and in 1985 by Černý [213]. Its main idea is to find a global minimum of a specific objective function attempting to escape local minima in the search process. Due to low complexity, it can be used in many various optimization problems, not only related to the VRPs. This method takes its name from the annealing in metallurgy, involving heating a solid material to a certain temperature at which it becomes liquid, followed by slow and controlled cooling down until the solid state is reached again and the metal particles are rearranged in a minimal energy molecular structure. An example of simulated annealing procedure is shown in Fig. 4.7.

The SA is a stochastic algorithm involving asymptotic convergence and allowing random movements in the searched neighborhood in order to escape local minima [1]. Although proposed more than 40 years ago, it still attracts some attention and is broadly used in many existing solutions for different variants of the vehicle routing problems.

Woch and Łebkowski [212] presented in 2009 a standard sequential simulated annealing algorithm applied to the VRPTW, which gave two new world's best solutions to Solomon benchmark set. In 2013, Afifi et al. [4] developed a simulated annealing algorithm for the VRPTW with synchronization constraints incorporating several local search techniques to deal with this problem. By implementing it the researchers were able to produce high-quality solutions in very short computational time detecting some new world's best solutions too.

Yu et al. [207] suggested in 2016 a simulated annealing heuristic for the hybrid vehicle routing problem (HVRP), an extension of the Green Vehicle Routing Problem. They focused on vehicles that use a hybrid power source, and the model considered the utilization of electric and fuel power depending on the availability of either electric charging or fuel stations. By developing SA with restart strategy combined with Boltzmann and Cauchy functions for determining the acceptance probability of a worse solution, the researchers were able to effectively solve the HVRP test instances.

In 2017, Wei et al. [198] proposed a simulated annealing algorithm for the CVRP with two-dimensional loading constraints combining mechanism of repeatedly cooling and rising the temperature with a new open space-based packing method. The computational results proved that such combination of approaches allows for reaching or improving best-known solutions for most instances.

Just recently in 2018, Linan-Garcia et al. [50] proposed another multi-phases metaheuristic algorithm based on Simulated Annealing to solve the CVRP with stochastic demands. Their algorithm comprises very custom phases of annealing including Fast Quenching Phase, the Annealing Boltzmann Phase, the Bose–Einstein Annealing Phase, and the Dynamical Equilibrium Phase applied in different ranges of temperature in the Simulated Annealing.

The SA was also used to solve pickup and delivery problems. The SA algorithm developed by Wang et al. [194] in 2013 for the VRPTW with simultaneous pickup-delivery, which also proved to be an effective metaheuristic due to results of computational tests on Wang and Chen benchmark set. There exists also a combination of the SA with the other heuristics such as Two-Stage Hybrid Local Search algorithm for the VRPTW proposed in 2001 by Bent and Hentenryck [15]. Their algorithm used the simulated annealing method to minimize the number of vehicles in the first phase of the VRPTW, whereas the large neighborhood search (LNS, see Section 4.2.5 for more details) was used to minimize total travel cost.

4.2.2 Tabu Search

Tabu Search (TS) was introduced and formalized by Glover [65] in 1959 as a metaheuristic search technique comprising local search methods and memory structures called tabu-list. Its main idea is to avoid cycling by inserting recently checked solution on the tabu-list, so during the search process, the solutions marked with tabu label are not taken into consideration. Such approach helps in getting out from the local minima and increases chances to find global, optimal solution. An example of tabu search method is shown in Fig. 4.8.

The initial implementations of the tabu search method to the routing problems were conducted in 1990s by Taillard [179] and Gendreau et al. [63] for the

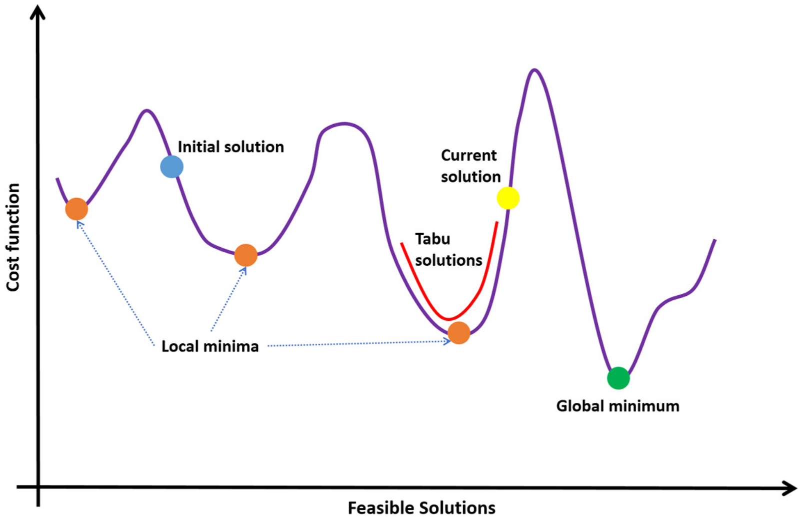

FIGURE 4.8 Tabu Search method (blue (mid gray in print version) dot – initial solution, orange (light gray in print version) dot – current solution, orange (light gray in print version) dot – local minimum, green (dark gray in print version) dot – global minimum, red (dark gray in print version) curve – tabu solutions).

CVRP and by Semet and Taillard [169] and Potvin et al. [152] for the VRPTW. The TS algorithm by Taillard [179] for the CVRP is still considered one of the best methods to solve it, and its main ideas comprise random tabu durations combined with continuous diversification mechanism penalizing frequently performed moves, making broader exploration of the search space. In turn, the tabu search algorithm by Gendreau et al. [63] is based on moving in each iteration a customer from one route to another where at least one neighbor is present. The consecutive insertions are carried out simultaneously with a local route improvement based on the GENI-exchanges (see Section 4.1.2 for more details). Similarly to Taillard's algorithm, infeasible solutions are also penalized, and random tabu durations are combined with continuous diversification mechanism.

Based on the initial tabu search algorithms, many other researchers conducted extensive studies and enhancements producing many different versions of the TS method applied to probably all variants of the vehicle routing problems. A good review on various tabu search heuristics for the VRPTW was given by Bräysy and Gendreau [26] in 2002. In 2004, Ho and Haugland proposed tabu search heuristics for the VRPTW with Split Deliveries, a variant where customer demands exceed vehicle capacity. Cordeau and Laporte [37] gave in 2005 a very good review on the most important heuristics for the whole family of the VRPs focusing on short/long memory structures, neighborhood structures, intensification, and comparison of computational results of various algorithms. A combination of the Tabu Search with guided local search principles were proposed in 2009 by Zachariadis et al. [208] for the VRP with two-dimensional loading constraints, and its implementation led to several new best known solutions. Moccia et al. [117] described in 2010 the incremental tabu search algorithm for the Generalized VRPTW introducing an incremental procedure to compute successive neighborhoods and demonstrating through extensive computational experiments that such general algorithm is competitive with other specialized heuristics for the VRPTWs. Brandao [23] reported in 2011 four new world's best solutions for the heterogeneous fixed fleet VRP by implementing a tabu search algorithm for this variant of the VRP.

More recently in 2012, Tarantilis et al. [184] proposed a tabu search solution framework for the Consistent VRP adopting a two-level master–slave decomposition scheme. The tabu search heuristic approach was employed on a dual mode basis to modify both the template routes and current daily schedules proving the competitiveness of such an approach by computational experiments on benchmark datasets. In 2013, Nguyen et al. [129] described a novel tabu search heuristic for the Time-dependent Multizone Multitrip VRPTW, where two types of neighborhoods related to the two sets of decisions of the problem together with an approach controlling the selection of the neighborhood type for particular phases of the search are able to combine exploration and exploitation capabilities for the search. Moreover, to drive the search to potentially unexplored good regions, the diversification strategy, guided by an elite solution set, is used. Another, very competitive iterated tabu search heuristic for the multi-compartment VRP was introduced in 2017 by Silvestrin and Ritt [171], which dominated all existing approaches in nearly all cases for this variant of the VRP.

An attempt to combine tabu search heuristics with simulated annealing was suggested by Wang et al. [196] in 2017 for a vehicle routing problem with cross docks and split deliveries by introducing a constructive heuristics with two layers. The first layer was used to optimize the allocation of trucks to cross docks, whereas the second layer was used to optimize the visitation order to suppliers and retailers for trucks assigned to each cross dock. Another interesting combination of tabu search with ejection chains for the multidepot Open VRP was proposed also in 2017 by Soto et al. [174]. They hybridized this approach also with multiple neighborhood search being able to generate neighborhoods from path moves and ejection chains, and by numerical and statistical tests they proved that such a combination outperforms state-of-the-art methods.

4.2.3 Adaptive Memory Procedures

The Adaptive Memory Procedure (AMP) was introduced in 1995 by Rochat and Taillard [159] and is based on the idea of the so-called *adaptive memory* being pool of good-quality solutions. The solutions in *adaptive memory* are always replaced with the new better-quality solutions coming from recombination of existing ones. The AMP by Rochat and Taillard was initially developed for the VRP. In this context, they proposed the process of extracting nonoverlapping

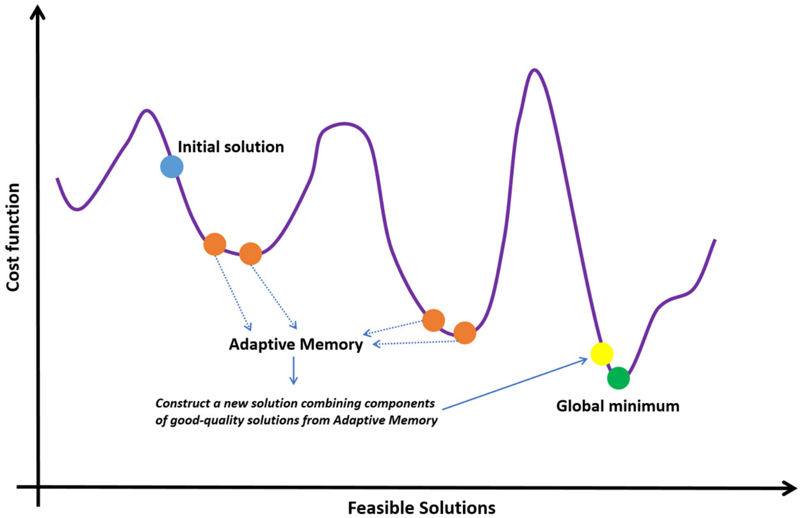

FIGURE 4.9 Adaptive Memory Procedure (blue (mid gray in print version) dot - initial solution, orange (light gray in print version) dots - good quality solutions, orange (light gray in print version) dot - new solution combining components of good-quality solutions from Adaptive Memory, green (dark gray in print version) dot - global minimum).

routes from the solutions followed by insertions of unrouted customers to create new feasible offsprings. Each time new offspring solution has a better quality than the worst one in the adaptive memory, and then such a better offspring solution replaces the worst solution in the adaptive memory. The AMP is illustrated in Fig. 4.9.

Bozkaya et al. [22] developed in 2003 an interesting combination of the adaptive memory procedure with tabu search algorithm for political districting optimization problem, in which the objective is to partition a territory into electoral constituencies subject to contiguity, population equality, and compactness side constraints. The authors performed the computational experiments on the real example of Edmonton city.

Tarantilis and Kiranoudis [183] modified in 2002 the original AMP by changing initial solution generation with the Clarke–Wright savings method combined with 2-Opt local moves, customer swaps between routes, and reinsertions of customers. Instead of extracting full routes from the adaptive memory as in original AMP technique by Rochat and Taillard, they suggested extracting particular route segments (so-called bone-routes) to generate new offspring solutions. The bone-routes proved to be very efficient by executing computational experiments over benchmark sets of the Capacitated VRP. This method was further improved in 2005 by Tarantilis [182] by introducing Solutions Elite Parts Search (SEPAS), an iterative method generating initial solutions by a systematic diversification technique and storing them in the adaptive memory. Generation of the new solutions was carried out using the tabu search heuristics, and such combination allowed the author to find several new best known solutions for the CVRP. The AMP was also applied to the other variants of the VRP such as VRP with multiple trips by Olivera and Viera [132] in 2007 or for the heterogeneous fixed fleet VRP by Li et al. [105] in 2010. In the latter one, Li et al. proposed a multistart adaptive memory programming (MAMP) and path relinking to solve this variant of the VRP. At each iteration, the MAMP was used to construct multiple temporary solutions, further improved by modified tabu search heuristics integrated with path relinking methods.

For the Fleet Size and Mix VRPTW variant, Repoussis and Tarantilis [157] proposed in 2010 an interesting AMP solution utilizing the basic concept of AMP framework combined with probabilistic semiparallel construction heuristic, a novel solution reconstruction mechanism, an innovative Iterated Tabu Search algorithm tuned for intensification local search and frequency-based long-term memory structures. The authors proved to improve the best reported cumulative and mean results for almost all problem instances in reasonable computational time.

More recently, in 2016, Gounaris et al. [72] presented the AMP method to solve the Robust CVRP under demand uncertainty. By focusing on verifying the robust feasibility of candidate routes and uncertainty sets, the authors were able to obtain new best known solution to 123 benchmark instances.

4.2.4 Variable Neighborhood Search

The Variable Neighborhood Search (VNS) was proposed in 1997 by Mladenovic and Hansen [115], which can be described as a framework for building heuristics exploiting the neighborhoods for both finding local optima and getting out of them by perturbation moves. Contrary to classical local search methods, VNS is not based on following the trajectory but rather on exploring increasingly distant neighborhoods of a given solution and moving to the new one only in case of improvement. Such a method usually leads to maintaining best characteristics of current solution and helps obtain neighboring solutions of better quality. After VNS was proposed, a lot of other researchers used this idea to combine VNS with other techniques, usually to create hybrids of VNS with tabu search and other improvement heuristics.

In 2003, Braysy [24] presented a novel Reactive VNS method for the VRPTW based on the modification of the original procedure combined with a new four-phase approach. In the first phase an initial solution is created using specialized route-construction heuristic, and a route-elimination method is applied in the second phase to further decrease the number of routes. The total traveled distance is minimized in the third phase by four new local-search procedures, and finally in the fourth phase the best solution is enhanced by modifying the objective function to escape from a local minimum.

The first attempt to solve the Multi-Depot VRPTW by a Variable Neighborhood Search method was undertaken by Polacek et al. [148] in 2004, and it was proven to be more effective than the baseline Tabu Search algorithm for this variant of the VRPTW.

Paraskevopoulos et al. [144] described in 2008 a very interesting solution methodology for the heterogeneous fleet VRPTW. Their approach was based on the idea of two-phase solution framework built upon a hybridized Tabu Search within a new Reactive Variable Neighborhood Search metaheuristic algorithm. Computational experiments proved the effectiveness of the approach and its applicability to realistic routing problems.

The VNS method was also applied to very large-scale vehicle routing problems, in particular, by Kytojoki et al. [101] in 2007, who applied it to the CVRP. The VNS procedure was used to guide standard improvement heuristics, and a strategy reminiscent of the guided local search metaheuristic was applied to help escape local minima. The authors proved that such a method is very robust, and by performing computational experiments they were able to find high-quality solutions for the problem instances with up to twenty thousands customers, which is considered very large scale in the CVRP.

More recently, in 2015, Wei et al. [199] presented a VNS for the CVRP with two-dimensional loading constraints, in which customer demand is a set of two-dimensional rectangular weighted items. Apart from the VNS to address the routing aspect, they used also a skyline heuristic to examine the loading constraints. Moreover, they utilized Trie data structure to speed up the search process to record the loading feasibility information of routes and to control the computational effort of the skyline spending on the same route. Based on the results of the extensive computational experiments, the authors proved that the proposed method outperformed all existing methods at that time for this variant of the CVRP.

4.2.5 Large Neighborhood Search

The Large Neighborhood Search (LNS) was introduced in 1998 by Shaw [170], initially for the VRPTW, and can be described as an iterative way of destroying and repairing the solution in the neighborhood. Destroying methods have some randomness such that different parts of the solution are destroyed and broader parts of the search tree are visited, and thus the searched neighborhood is larger than in classical local search methods. The removed customers from routes are usually reinserted back using Constraint Programming; however, other methods can be used too. The Constraint Programming strategies are used to check feasibility of the moves, to handle side constraints, and to allow iterative improvements in the search procedures.

Ergun et al. [49] have extended this concept further by proposing Very Large Neighborhood Search (VLNS) for the VRP by introducing operation on multiple routes simultaneously, which allowed even broader search. This advantage comes however with much higher effort required to perform extensive repairing moves in each iteration.

In 2006, Ropke and Pisinger [160] extended the idea of the LNS to the Adaptive Large Neighborhood Search (ALNS) for the Pickup and Delivery Problem with Time Windows. In their concept, multiple destroy and repair methods were allowed within the same search iteration. Another adaptive LNS heuristic but for Two-Echelon VRP (2E-VRP) was proposed in 2012 by Hemmelmayr et al. [75], who proposed new neighborhood search operators by exploiting the structure of the problem, and these operators were used in a hierarchical scheme reflecting the multilevel nature of the problem. Ribeiro and Laporte [158] in 2012 also developed an adaptive LNS for the Cumulative Capacitated VRP, which is a variation of the classical CVRP where the objective is to minimize the sum of arrival times at customers, instead of the total routing cost. The VRP with multiple routes was solved by an adaptive LNS by Azi et al. [8] in 2014, who developed various destruction and reconstruction operators working either at the customer, route, or workday level.

More recently, in 2016, Grangier et al. [73] developed an adaptive LNS for the Two-Echelon Multi-Trip VRP with satellite synchronization, addressing constraints arising in city logistics such as time window constraints, synchronization constraints, and multiple trips. They suggested adaptive LNS combined with custom destruction and repair heuristics and efficient feasibility checks for moves.

4.2.6 Greedy Randomized Adaptive Search Procedure

The Greedy Randomized Adaptive Search Procedure (GRASP) was introduced by Feo and Resende [51] in 1995 and is based on iterative two-phase search algorithm comprising construction and local search phases applied to various combinatorial optimization problems. In each iteration in the construction phase, a feasible solution is constructed by a randomized greedy function. The solution is then iteratively improved in the second phase by local search movements. An illustration of the GRASP is shown in Fig. 4.10.

The GRASP method was applied to the VRPTW by Kontoravdis and Bart [95] in 1995, and by developing additionally three lower bounding heuristics, they found optimum solutions for almost all test instances up to 100 customers. GRASP is also very often combined with evolutionary local search (ELS) methods such as VRP heuristic by Prins [154] from 2009, who proposed solutions encoded as giant tours and a robust local search based on a sequential decomposition of moves. Based on the experimental studies, their approach outperformed all previous methods at that time except Active Guided Evolution Strategy (AGES) algorithm by Mester and Bräysy (see Section 4.2.16 for more details). In 2010, Duhamel et al. [47] addressed the GRASP-ELS approach to the Capacitated Location Routing Problem (CLRP), which is defined by a set of depot locations with opening costs and limited capacities, a homogeneous fleet of vehicles, and a set of customers with known demands. The objective of the CLRP is to assign customers to opened depots and to design vehicle routes in order

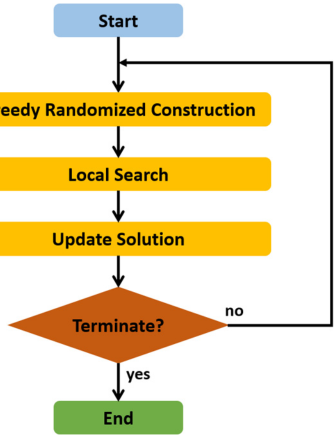

FIGURE 4.10 Illustration of the Greedy Randomized Adaptive Search Procedure (GRASP).

to minimize both the cost of open depots and the total cost of the routes. Their method comprises GRASP with ELS combined with searching within solutions identified by giant tours without trip delimiters. Giant tours were evaluated by a splitting procedure minimizing the total cost due to vehicle capacity, fleet size, and depot capacities.

The combination of GRASP with Variable Neighborhood Search (VNS) (see Section 4.2.4 for more details) and path relinking methods was proposed by Villegas et al. [193] in 2011 to solve the Truck and Trailer Routing Problem (TTRP). The TTRP comprises a heterogeneous fleet composed of trucks and trailers serving a set of customers, some only accessible by truck and others accessible with a truck pulling a trailer.

In 2012, Marinakis [109] developed a modified version of GRASP, called Multiple Phase Neighborhood Search-GRASP (MPNS-GRASP) for solving the Capacitated VRP. He utilized a stopping criterion based on Lagrangean Relaxation and Subgradient Optimization in addition to the Circle Restricted Local Search Moves strategy being a novel way for expanding the neighborhood search.

4.2.7 Particle Swarm Optimization

The Particle Swarm Optimization (PSO) introduced in 1995 by Kennedy and Eberhart [92] aimed at producing computational intelligence by exploiting analogues of social interaction rather than individual cognitive abilities [149]. It is an optimization method based on the idea of iterative improvements of a solution with regard to certain quality measures. PSO is using a population of particles representing solutions to move them around in the search space due to some mathematical rules over particles velocity and position. All particles movements are guided toward best known positions in the search space, thus moving the whole swarm toward the global minimum [149]. Similarly to flock of birds collectively searching for food, the swarm is likely to move close to an optimum of the fitness function. An illustration of the Particle Swarm Optimization is shown in Fig. 4.11.

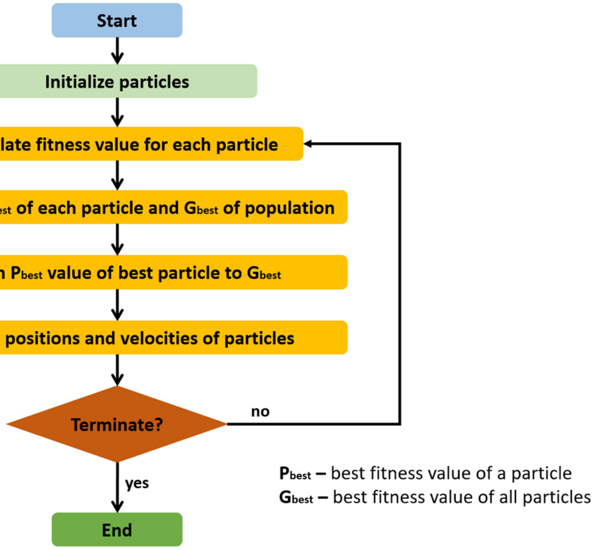

FIGURE 4.11 Illustration of the Particle Swarm Optimization.

The PSO was originally designed for solving continuous optimization problems, but there was created also a mechanism to tackle discrete optimization problems, the so-called discrete PSO [200]. This discrete PSO method was taken into account by many researchers for different variants of the vehicle routing problems. In each PSO algorithm for the VRP variants, most important questions are how to represent a solution by a particle and how to build an efficient decoding method to decode particle back into a solution.

The discrete PSO was used initially in 2004 by Yang et al. [200] to solve the CVRP where particles were represented as an array of dimensions and positions related to customers served by particular vehicles.

Ai and Kachitvichyanukul [5] developed in 2009 a PSO algorithm with multiple social structures to solve the VRP with simultaneous pickup and delivery. They proposed a random key-based representation of solution and corresponding decoding method. The decoding method was used to transfer the particle to a priority list of customers and to a priority matrix of vehicles to serve all customers. The routes were then constructed based on the customer priority list and vehicle priority matrix. The authors proved the competitiveness of such approach and were able to find some new best known solutions.

In 2010, Marinakis et al. [110] developed an interesting combination of the PSO algorithm with the multiple phase neighborhood search-greedy randomized adaptive search procedure (MPNS-GRASP), expanding the neighborhood search strategy and path relinking to solve the classical VRP. It proved to be suitable and effective for solving very large-scale problems within short computational time and was ranked in the fifth place among the 39 most known and effective algorithms in the literature and in the first place among all nature inspired methods for the VRP.

In 2011, MirHassani and Abolghasemi [114] developed a PSO algorithm for the Open VRP, where vehicles do not return to depot after serving last customer in the route. They proposed a particular decoding method in which the customer position vector is generated in descending order and all customers are assigned to a route always taking into account feasibility constraints. The extra one-point moves were also applied on constructed routes, which seems promising in achieving better solutions.

A combination of the PSO algorithm with a genetic algorithm to solve the Capacitated VRP with fuzzy demands was developed in 2012 by Kuo et al. [100]. This algorithm used the idea of a particle best solution and the best global solution combined with crossover and mutation of Genetic Algorithm. The modification of particle coding was also performed to ensure that particle always generates a new feasible solution.

In 2013, Goksal et al. [68] developed a very interesting combination of the PSO with Variable Neighborhood Search to solve the VRP with simultaneous pickup and delivery. Moreover, they introduced a solution represented by a giant tour without trip delimiters, and an annealing-like strategy was applied to preserve the swarm diversity. Their algorithm was able to improve 104 best known solutions for this variant of the VRP.

An interesting solution for particle coding and decoding in the PSO algorithm was also proposed by Ho et al. [81] in their hybrid chaos-particle PSO algorithm for the VRPTW. The chaos algorithm was employed to reinitialize the particle swarm, and an insertion heuristic algorithm was incorporated to build the feasible vehicle routes in the particle decoding process. Moreover, a particle swarm premature convergence judgment mechanism was combined with Gaussian mutation into the hybrid chaos algorithm and used when the particle swarm falls into the local convergence.

More recently, in 2016, Yao et al. [205] proposed an improved PSO algorithm for carton heterogeneous VRP with a collection depot utilizing a self-adaptive inertia weight and a local search strategy. The computational results showed that developed model is feasible with savings of about 28% in total delivery cost.

4.2.8 Ant Colony Algorithms

The Ant Colony Algorithm (ACO) was first introduced in 1992 by Dorigo in his PhD thesis [44] and was based on the behavior of ants seeking paths between their colony and sources of food, laying some pheromone on trails. The fact that

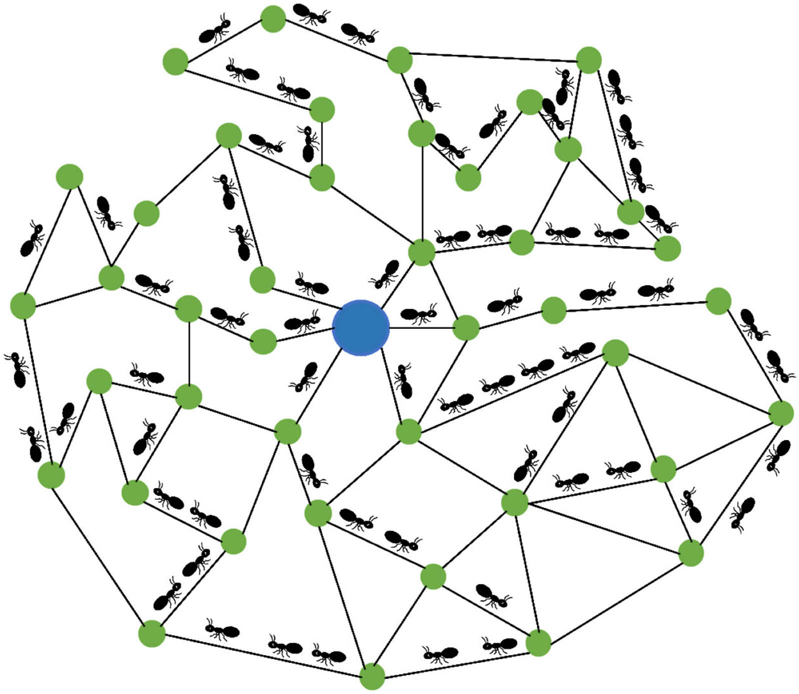

FIGURE 4.12 Illustration of the Ant Colony algorithm (ants seeking paths between their colony (blue (dark gray in print version) dot) and sources of food (green (mid gray in print version) dots), laying some pheromone on trails with information of quantity and quality of food; ants always follow the same path, which is the shortest path).

ants always follow the same path, which is indeed the shortest path, was the main motivation to take advantage of such real natural behavior of ants. The walking ants mark trails by laying down pheromones with information of quantity and quality of food. Such idea could be translated to the vehicle routing problems as searching in the neighborhood for good-quality solutions. An illustration of the Ant Colony Algorithm is shown in Fig. 4.12.

It was initially used to solve TSP problem instances by Colorni et al. [36] and then designed for the VRP and CVRP in 1999 by Bullnheimer et al. [28]. The Ant Colony method was evolving and the so-called *reinforcement learning* mechanism emerged there meaning automatic adjustments of heuristic components as the search process evolves [188]. This idea incorporates learning mechanism how to make proper decisions based on behavior and introducing reinforcement methods to improve the quality of solutions.

In the next years, the Ant Colony algorithms became very popular methods for the other variants of the VRP. In 1999, Gambardella et al. [58] proposed an ant colony optimization algorithm for the VRPTW with an idea of association the attractiveness measure with the arcs. They treated artificial ants as parallel processes, and their role was to create feasible solutions. Two ant colonies were used, the first for minimizing the number of routes and the second for minimizing the total distance traveled. Similarly to natural behavior of ants, also such artificial ants were cooperating by exchanging information about solutions through pheromone updates. Each time a new solution with smaller number of routes or smaller total distance was found, such an information was spread out to other ants through pheromones. Baran and Schaerer [12] further improved this algorithm in 2003 by developing only one colony to get a set of Pareto optimal solutions considering three objectives at the same time, the number of vehicles, the total traveling time, and the total delivery time. The computational experiments proved that the new technique outperforms the original approach. Modification to the original ACO was tackled also by Bell and McMullen [14] in 2004, who focused on the size of the candidate lists used within the algorithm, which appeared a significant factor in finding improved solutions.

In 2008, Donati et al. [43] developed multi-ant colony systems for the Time-Dependent VRP by combining the original ACO approach with discretizing the time space in a suitable number of subspaces. Moreover, they introduced new time dependent local search procedures and conditions guaranteeing that feasible moves are checked in constant time. Such a model was also integrated with a robust shortest path algorithm to compute time-dependent paths between each customer pairs.

The ACO approaches were further investigated and enhanced by Fuellerer et al. [55] in 2009 for the two-dimensional loading VRP, excellent behavior of which was proven through extensive computational results. Their algorithm was flexible enough in handling different loading constraints, including items rotation and rear loading so that it allowed for qualitative conclusions of practical interest in transportation, such as evaluating the potential savings by more flexible loading configurations. Moreover, it was one of the first ACO methods successfully combining two totally different heuristic measures of loading and routing within one pheromone matrix. Based on the computational results, their algorithm was able to reach optimal solutions on small-size instances, and for larger-size instances, it clearly outperformed all other heuristics at that time.

In 2009, Gajpal and Abad [57] took advantages of the ACO for solving VRP with simultaneous pickup and delivery, getting very competitive results with the other approaches. They used construction rules and two multiroute local search schemes to solve this variant of the VRP. It is remarkable to note that their version was able to solve also the other VRP with backhaul and mixed load.

More recently, in 2014, Reed et al. [156] developed the ACO algorithm for the multicompartment VRP associated for example with a collection of recycling waste from households, treated as nodes in a spatial network. They introduced preprocessing by k-means clustering, greatly reducing the computation time and producing improved routings for networks where the nodes are concentrated in separate clusters. The algorithm was also extended to model the use of multicompartment vehicles with kerbside sorting of waste into separate compartments for each category. Abdulkader et al. [2] also tackled the multicompartment VRP in 2015 creating a hybridized version of the ant colony algorithm. They combined the local search method with existing ACO algorithm and proved by computational experiments that the new approach gives better results.

Kalayci and Kaya [90] empowered in 2016 Variable Neighborhood Search (see Section 4.2.4 for more details) method with the ACO resulting in a very competitive algorithm to solve the VRP with simultaneous pickup and delivery. The application of the VNS provided intensive local search, and its weakness of lack of memory structure was minimized by utilizing long-term memory structure of Ant Colony System, and the overall performance of the algorithm was boosted. It is remarkable to note that in the proposed algorithm, instead of ants, the pheromones were released on edges. The ants provide a perturbation mechanism for the integrated algorithm using the pheromone information to explore search space broader and jump out from local minima. Numerical experiments proved that such a combination of the VNS with ACO is very robust and efficient in terms of both quality and CPU time, and the authors reported some new best known solutions too.

In 2018, Goel and Maini [67] developed a very interesting hybridization of the Ant Colony System with firefly algorithms (HAFA) for solving the vehicle routing problems. The ACS provided the basic framework to proposed algorithm, and Firefly Algorithm (FA) was used to search for the unexplored solution space. Moreover, a novel pheromone shaking technique was incorporated in ACS to escape local minima due to avoiding pheromone stagnation on the exploited regions. The computational experiments proved that such combination outperforms other FA-based approaches in terms of convergence time.

4.2.9 Artificial Bee Colony Algorithms

The Artificial Bee Colony (ABC) algorithm was developed in 2005 by Karaboga [91]. Similarly to other swarm optimization algorithms, it is also based on two fundamental concepts, namely self-organization and division of labor, which allow problem solving systems to self-organize and adapt to the environment. The ABC model related to collective intelligence of honey bees comprises employed and unemployed bees and food sources [91]. It defines also recruitment to a nectar source and abandonment of such source as two leading methods of the honey bees behavior. The value of the food source for honey bees usually depends on its richness, proximity to the nest, and the ease of forage extracting. The employed bees are associated with a certain food source they are exploiting, and they share the information about such source with other bees. In turn, unemployed bees are constantly searching the neighborhood of the nest for new sources of forage and finally establishing new directions by gathering all information together shared by all the bees [91]. The information exchange among bees is happening in the so-called dancing area, and dance is called a waggle

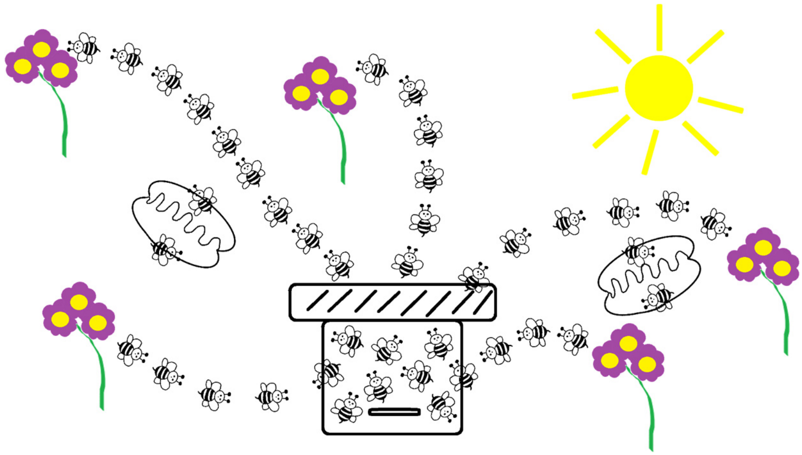

FIGURE 4.13 Illustration of the Artificial Bee Colony algorithm (the value of the food source (flowers) depends on its richness, proximity to the nest, and the ease of forage extracting and can be associated with good solutions; the information exchange among bees is happening in the so-called waggle dance (illustrated as two dancing bees in a circle).

dance. Based on that, the profitability of new food sources is judged and chosen. An illustration of the Artificial Bee Colony algorithm is shown in Fig. 4.13.

The ABC algorithm by Karaboga simulates both behavior of honey bees and foraging in order to solve potential optimization problems. A possible solution is represented by a food source, whereas the amount of food in the source is related to the solution quality. The search process is associated with selecting various sources of food and abandoning them if the solution represented by the food source is not improved for a certain number of trials [91].

In the field of swarm intelligence the ABC algorithms and Bee Colony Optimization (BCO) methods have much less attention than the other methods such as Ant Colony Optimization mostly because of more complex implementation; however, the obtained results are still competitive and worth further investigations. In 2012, Bhagade et al. [18] developed the ABC algorithm for the classical TSP enhancing it additionally by using the nearest-neighbor method. The effectiveness of the bee paths was evaluated with tour length and bee travel time. Based on the computational results, the authors proved that the ABC algorithm can be efficiently used for solving the TSP and moreover is highly flexible and can be extended to other optimization problems by considering relatively few control parameters.

The ABC algorithm was also taken into account for solving the Capacitated VRP by Szeto et al. [176] in 2011 and for solving the Periodic VRP by Yao et al. [203] in 2013. To improve the performance of the ABC algorithm, Yao et al. combined it with multidimensional heuristic information and a local optimization based on a scanning strategy. By running numeric experiments they proved that the proposed modification of the ABC algorithm is a powerful method for solving the PVRP.

Zhang et al. [209] developed in 2014 a hybrid artificial bee colony algorithm with hybrid operators for the Environmental VRP. The performance of this approach was evaluated through computational tests on CVRP instances, and the results showed that a hybrid approach outperforms the original ABC algorithm by 5% on average. An interesting combination of the adaptive memory with the ABC algorithm for the Green VRP with cross-docking was proposed in 2016 by Yin et al. [206]. Their hybrid metaheuristic was able to reach higher fuel efficiency than the tabu search algorithm by managing the loading along the route and yielding less total cost. Such a method proved to be robust against the problem size, and convergence of the objective value was guaranteed with high confidence.

The application of the ABC algorithm for the VRPTW was undertaken by Alzaqebach et al. [6] in 2016, who proposed a modified version of the ABC tackling the issue that high exploration ability of the ABC usually slows down its convergence speed. In their approach a list of abandoned solutions is used by the scout bees to memorize the abandoned solutions. The scout bees select then a random solution from the list and replace by a new solution with random routes chosen from the best solution. Yao et al. [204] also used in 2017 a modification of the ABC algorithm to solve the VRPTW. They proposed the enhancements of a local optimization based on a crossover operation and a scanning strategy.

Just recently, Sedighizadeh and Mazaheripour [168] developed in 2017 and optimization technique of multiobjective VRP using a hybrid algorithm based on Particle Swarm Optimization combined with ABC algorithm considering precedence constraints. In particular, they proposed a solver algorithm in which the idea is to consider different constraints of the problem, use penalty method, and additional segmentation constraint methods to obtain the best vehicle routes.

Ng et al. [128] also developed (very recently in 2017) a multiple-colony ABC algorithm for the CVRP and rerouting strategies under time-dependent traffic congestion. They took into account a rerouting strategy to solve the problem of inefficient vehicle routing caused by traffic congestion and proposed a flexible delivery rerouting strategy aiming at reducing the risk of late delivery. They introduced a novel scheme using a Multiple Colonies Artificial Bee Colony algorithm to solve the problem of solutions trapped in local minima. Such a design of the outstanding bee selection for colony communication proved to be superior in exploitation.

4.2.10 Bat Algorithms

The Bat Algorithm (BA) is a relatively new bioinspired metaheuristic introduced in 2010 by Xin-She Yang [202] and is based on echolocation behavior of microbats when searching for their prey in nature [177]. An illustration of the Bat algorithm is shown in Fig. 4.14.

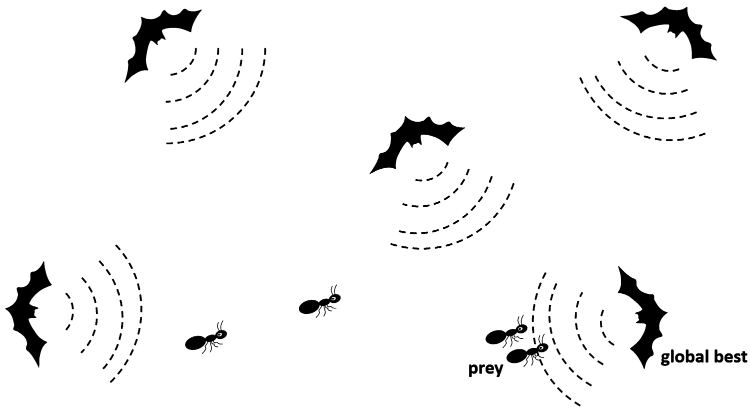

FIGURE 4.14 Illustration of the Bat algorithm (microbats use echolocation when searching for their prey (insects); good-quality solutions are associated with bigger quantity and quality of prey).

Taha et al. [177] developed in 2010 a hybrid algorithm executing a discrete version of the bat algorithm combined with the LNS (see Section 4.2.5 for more details) to solve the VRPTW. Their algorithm aimed at improving the performance of the BA by the destroy and repair paradigm of the LNS, allowing bats to explore a large part of the solution space.

Many more Bat Algorithms for various variants of the routing problems were developed just in a few years. In 2016, Zhou et al. [211] proposed a hybrid BA with Path Relinking for the Capacitated VRP, which is based on the framework of the continuous bat algorithm, GRASP, and path relinking method built into the BA. Additionally, the random subsequences and single-point local search are operated with certain loudness to further improve the performance of the developed algorithm. Osaba et al. [138] proposed in 2017 the BA with random reinsertion operators to solve the VRPTW. This algorithm comprises both improved movement strategy and diverse heuristic operators to deal with VRPTW. They unified the search process and the minimization phases by using selective node extractions and subsequent reinsertions. In 2018, Wang et al. [195] presented a self-adaptive BA algorithm applied for the first time for the truck and trailer routing problem, a generalization of the VRP involving a group of geographically scattered customers served by the vehicle fleet including trucks and trailers. They developed the BA combined with a local search procedure performed by five different neighborhood structures. Additionally, a self-adaptive tuning strategy was implemented to preserve the swarm diversity.

4.2.11 Cuckoo search

The Cuckoo Search is a metaheuristic optimization algorithm introduced by Yang and Deb [7] in 2009 inspired by the cuckoo birds being the "brood parasites" birds. They never build they own nests and instead lay their eggs in nests

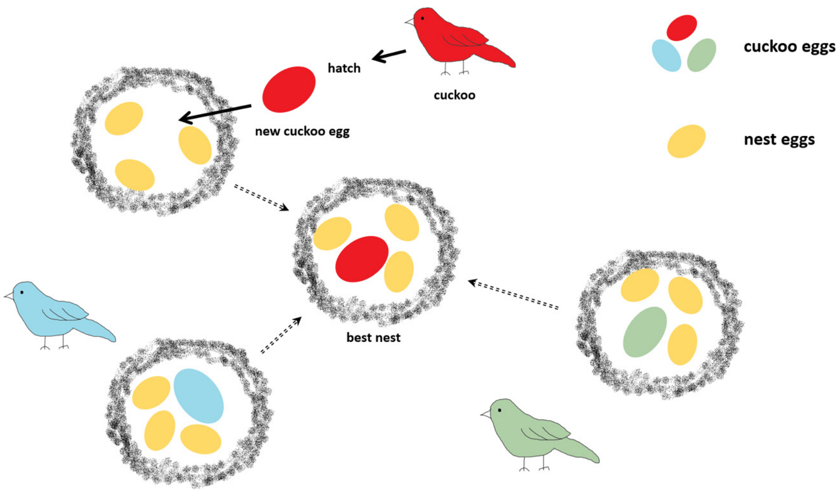

FIGURE 4.15 Illustration of the Cuckoo Search (orange (light gray in print version) – nest eggs, red/blue/green (dark/mid/light gray in print version) – cuckoo eggs; each egg represents a solution, and each cuckoo egg represents a potentially better solution; the Cuckoo Search algorithm is based on the idea of creating subsequent generations of nests containing best eggs and thus highest quality solutions).

of other host bird nests [86]. Moreover, if the host bird identifies eggs that are not theirs, then it throws them away from nest or leaves its nest and simply builds a new nest somewhere else. In terms of optimization algorithms, each egg represents a solution, and each cuckoo egg represents a new potentially better solution. The Cuckoo Search algorithm is based on the idea of creating subsequent generations of nests containing best eggs and thus highest quality solutions. In each iteration, each cuckoo lays one egg at a time and leaves it in a randomly selected nest, followed by selection of nests with highest quality eggs. Moreover, the number of host nests is fixed, and the cuckoo eggs are identified by host birds with a certain probability. In case cuckoo eggs are identified, the host bird either throws such solutions away or leaves its nest and builds a new nest. An illustration of the Cuckoo Search is shown in Fig. 4.15.

Initially, there were attempts to take advantage of the Cuckoo Search algorithm to the Traveling Salesman Problem. In 2012, Jati et al. [82] developed a discrete cuckoo search for TSP considering discrete step sizes and the cuckoo's updating scheme; however, the results showed that it was trapped into local minima for some TSP instances. Ouyang et al. [142] proposed in 2013 a novel discrete cuckoo search algorithm for Spherical TSP based on the Levy flight and brood parasitic behavior. They applied study, A, 3-Opt, and search-new-nest operators to speed up the convergence. An interesting combination of the Genetic Algorithm with Cuckoo Search and 2-Opt movements was presented in 2013 by Abu-Srhan and Al Daoud [3] to avoid the local minima problem and take advantage of the powerfulness of the GA, all applied to the TSP. More recently, in 2015 Ouaarab et al. [141] developed the random-key cuckoo search (RKCS) algorithm for solving the TSP by introducing a simplified random-key encoding scheme to pass from a continuous space (real numbers) to a combinatorial space. They also applied displacement of a solution in both spaces using Levy flights.

Zheng et al. [210] proposed an interesting combination of GRASP with Cuckoo Search applied to the Vehicle Routing Problem. In their version, called CS-GRASP (more information on GRASP can be found in Section 4.2.6), they utilized path relinking, swap, and inversion strategy. In 2016, Chen and Wang [33] developed a hybrid cuckoo search for the VRP based on the combination of Particle Swarm Optimization (PSO), Optical Optimization, and Cuckoo Search method. The Optical Optimization was used to initialize population with high-quality solutions optimized further by PSO. Cuckoo Search was then iteratively used to optimize the rest of the individual solutions. Teymourian et al. [185] developed in 2016 hybrid methods of Cuckoo Search combined with Local Search and Improved Intelligent Water Drops algorithm to control the balance between diversification and intensification of the search process.

4.2.12 Firefly Algorithms

The Firefly Algorithm (FA) was introduced in 2008 by Yang [201] inspired by flashing behavior of fireflies. The main purpose why fireflies flash is to signal and attract other flies. Yang introduced the FA assuming that all fireflies are unisexual, attractiveness is proportional to brightness, and that fireflies move randomly when there are no brighter fireflies in the neighborhood.

In 2011, Yati and Suyanto [83] developed an evolutionary discrete FA algorithm to solve the TSP by examining discrete distance between two fireflies and movement schemes. Their algorithm however without combination with the other methods was trapped in local minima in case of some TSP test instances. This idea was further explored by Kumbharana and Pandey [99] in 2013, who proposed the FA for the TSP comprising constructing a suitable conversion of the continuous functions of attractiveness, distance, and movement into new discrete functions. In 2017, Jie et al. [84] presented an improved version of the FA algorithm for the TSP by redefining the distance of FA by introducing a swap operator and a swap sequence to avoid falling into local minima and to accelerate convergence speed. They also adopted dynamic mechanism based on neighborhood search algorithm, proving by experiments that such approach is very competitive.

The initial research on the application of the FA for the routing problems was started just a few years ago by Pan et al. [143] in 2013, who developed the FA for the VRPTW. They introduced the coding and design of disturbance mechanism of elicit fireflies in their FA. In 2016, Osaba et al. [139] developed the first FA algorithm to solve the Rich VRP by introducing the Hamming Distance function to measure the distance between two fireflies of the swarm. Osaba et al.

[137] designed also an evolutionary discrete FA to the VRPTW by introducing some novel route optimization operators. These route optimization operators were used to incorporate minimizing the number of routes for a solution in the search process and to analyze all routes of the solution. They allowed them to increase the diversification capacity of the search process contrary to classical node and arc exchange-based operators.

4.2.13 Golden Ball Algorithms

The Golden Ball (GB) is a multipopulation metaheuristic based on soccer concept introduced very recently in 2014 by Osaba et al. [135]. In the initialization phase of the GB, the whole population of players (indicated as solutions) is created, followed by random division of players into fixed number of subpopulations called teams. Each team has its own couch related to training method, which is also randomly assigned to each time. The training method can be also associated with the way how each player evolves individually in the team, and in terms of the vehicle routing problems, it might be associated, for example, with 2-Opt moves. Osaba et al. introduced also the so-called Custom Training, which is a special training applied to the player trapped in local minimum and is based on a training with cooperation with the best player in the team, usually by Crossover operation. The second phase is called the competition phase and is divided into seasons and weeks. In each week, all teams train independently, and they face each other by creating a soccer league. Moreover, there are transfers at the end of each seasons, where both coaches and players can switch teams. An illustration of the Golden Ball Algorithm is shown in Fig. 4.16.

Initially, the GB was introduced for the TSP in 2014 by Osaba et al. [136], and it was proved to be competitive with the Evolutionary Simulated Annealing and Tabu Search approaches. This algorithm was later improved by Ruttanateerawichien et al. [163] to handle Capacitated VRP, and based on experiments, it was able to find new best known solutions for three benchmark instances. The random keys representation to encode solutions in the GB algorithm was proposed in 2015 by Sayoti and Riffi [166], who applied this technique for the TSP.

More recently, in 2016, Ruttanateerawichien et al. [162] introduced a new technique, where a team represents the CVRP solution, and the players represent routes in the team, whereas in the previous work the CVRP solution was modeled by a player. Additionally, the solution quality of players and teams was improved by intraroute and interroute improvement algorithms. In 2018, Guezouli et al. [74] designed efficient GB algorithm based on clustering to solve the Multi-Depot VRP with Time Windows. They proposed a different solution representation from the original one and embedding a clustering algorithm to solve the problem more efficiently.

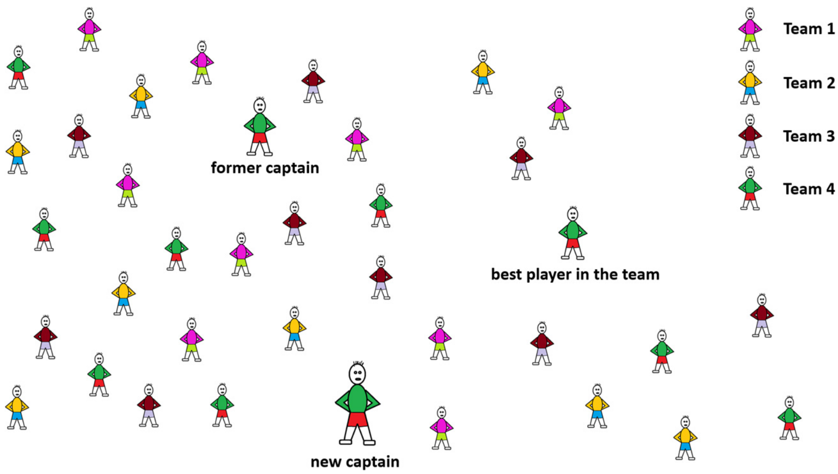

FIGURE 4.16 Illustration of the Golden Ball algorithm (each soccer represents a solution; soccers are divided into teams (populations); each team has its own captain (related to training method) and each player evolves individually in the team and may become the best player in the team (best solution in population).

4.2.14 Gravitational Search Algorithm

The Gravitational Search Algorithm (GSA) was introduced in 2009 by Rashedi et al. [155] based on the gravity law and mass interactions. The GSA was inspired by gravitation being the tendency of masses to accelerate toward each other, which is also one of the four fundamental interactions occurred in nature next to the electromagnetic force and the weak and strong nuclear forces [167]. Based on the fundamental Newton law of gravity, each particle attracts other particles with a gravitational force, which is directly proportional to the product of their masses and inversely proportional to the square of the distance between them [155]. In the GSA, agents are considered as objects, and their masses are related to their performance. All the agents attract each other by the gravity force communicating each other through gravitational force. The heavy objects—related to high-quality solutions—move slower than lighter objects, and this guarantees the exploitation step of the GSA algorithm [155]. Moreover, each agent can be characterized by position, inertial mass, and passive and active gravitational mass. The position of the agent is related with a solution, whereas the inertial and gravitational masses are determined by a fitness function so that the algorithm is navigated by properly adjusting these masses. It is expected in the GSA that during execution all the objects will be attracted by the heaviest one corresponding to a global minimum in the search space. An illustration of the Gravitational Search algorithm is shown in Fig. 4.17.

Chen et al. [32] proposed in 2011 a hybrid GSA algorithm for the TSP based on random-key encoding scheme of the solutions combined with simulated annealing. They incorporated into it also a multitype local improvement scheme

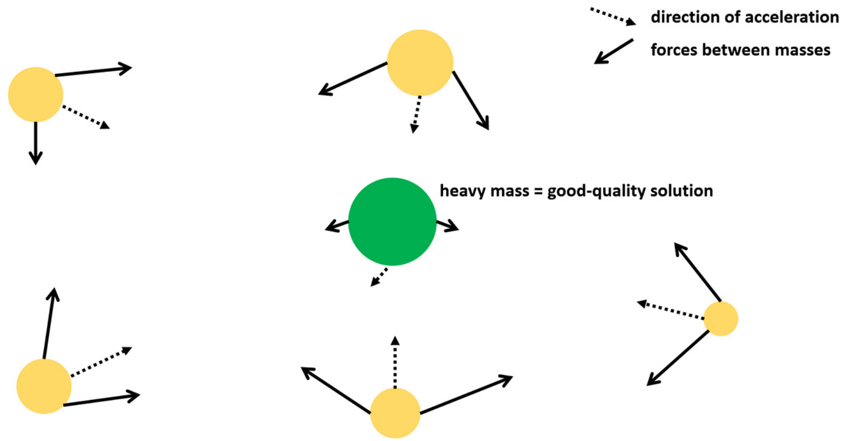

FIGURE 4.17 Illustration of the Gravitational Search Algorithm (bigger dots represent heavier masses, thus better solutions; solid arrows indicate forces between mases; dotted arrows indicate an acceleration direction).

used as a local search operator. The simulated Annealing method was utilized into proposed algorithm too, in order to manipulate the iteration progress algorithmically. Hosseinabadi et al. [80] presented in 2012 the GSA for the TSP using velocity and gravitational force in physics based on random search concepts. In 2014, Dowlatshahi et al. [45] developed a Discrete GSA for the TSP with path relinking strategy instead of original agent movements. Their algorithm was ranked ninth compared with 54 different algorithms for the TSP.

The GSA approach was also used to solve the Open VRP by Hosseinabadi et al. [79] in 2015. Their algorithm was based on random search concepts utilizing speed and gravity parameters, and the researcher agents were connected each other based on Newton's gravity and motion laws.

4.2.15 Bacterial Foraging Optimization Algorithm

The Bacterial Foraging Optimization Algorithm (BFOA) was introduced in 2002 by Passino [145] to mimic biological principles shown in the foraging behavior of E. coli bacteria for distributed optimization and control. E. coli bacteria live in intestines of most animals on the Earth, and they have a control system directing its behaviors in food foraging. The foraging process of E. coli bacteria comprises searching for food, deciding whether to enter possible food region or not, careful searching if to enter into a new area, and deciding whether to stay in actual region or move on to a new better area after consumption some food in the current region [76]. In the BFOA a swarm of bacteria is used as a searching agent for a solution to a specific optimization problem. A bacteria position represents a solution of the problem by a simple sequence of customer nodes. The bacteria direction vector is used to represent its ability to search for solution and is driving bacteria movement. In each step of the BFOA, all bacterium moves from one position to another are based on these directions, and by moving various solutions of the problem are evaluated.

The solution of the TSP using BFOA was carried out by Verma et al. [189], where the key step used in the algorithm was computation of chemotaxis, where a bacterium takes steps over the foraging landscape to reach the areas with high-nutrient food. The movement of a bacterium was led by computing direction and distance matrices, and additionally in case the bacterium does not reach a city within the minimum distance constraint in its first step, it searches for other cities in the next steps.

Much more BFOA algorithms were proposed for variants of the VRP family. In 2011, Hezer and Kara [76] applied for the first time the BFOA technique to solve the VRP with Simultaneous Delivery and Pick-up. Niu et al. [130] developed in 2012 the BFOA with adaptive chemotaxis step to solve the VRPTW, in which they applied additionally a nonlinearly decreasing exponential modulation model to further improve the efficiency of the algorithm.

In 2014, Tan et al. [180] proposed another BFOA for the VRPTW comprising two versions with linear and nonlinear decreasing chemotaxis steps to obtain the best solutions of a given VRPTW problem. Tan et al. [181] further improved their solution by developing adaptive comprehensive learning BFOA for the VRPTW to keep a good balance between the exploration and exploitation. They proposed also the comprehensive learning mechanism maintaining the diversity of the bacterial population and thus alleviating the premature convergence.

The solution of the Heterogeneous Fixed Fleet VRP was proposed in 2015 by Gan et al. [59], who developed a new method based on structure-redesign-based bacterial foraging optimization (SRBFO) combined with time decreasing chemotaxis step size mechanism. Additionally, the position of bacteria was encoded by $2N$ dimensions, where the first N dimensional vectors indicated the corresponding vehicle, and the next N dimensional vectors were related to the execution order of the corresponding vehicle routing.

In 2017, Li et al. [104] proposed the BFOA for the VRP with Pickup and Delivery. They established the mathematical model aiming at minimizing the total travel time and the total, which was combined with dynamic variable step factor and propagation threshold and death threshold to copy the best individuals and eliminate the inferior individuals.

4.2.16 Genetic and Evolutionary Algorithms

The Genetic Algorithms (GAs) are optimization search methods based on the evolution process in nature, and they imitate the biological process of natural selection where stronger populations among different species survive [69]. The main concepts of GAs were developed by Holland [78] in the 1960s and 1970s, and practicality of using GAs for solving complex problems was elaborated later in 1975 by De Jong [42] and in 1989 by Golberg [69]. The GAs

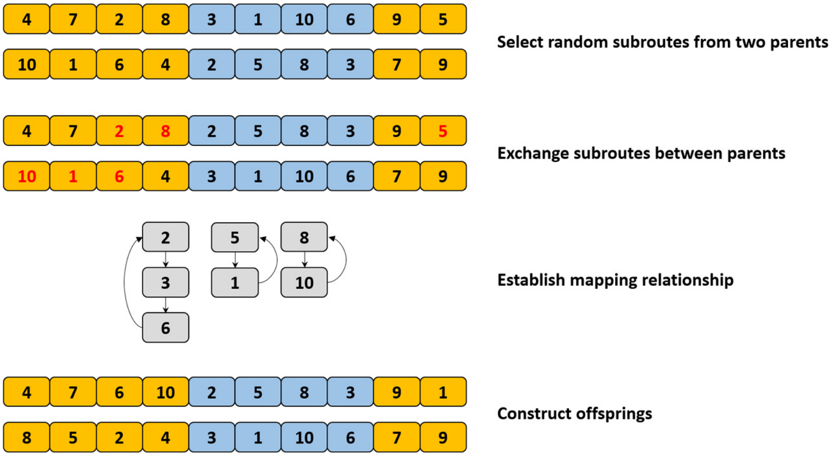

FIGURE 4.18 Illustration of the PMX (Partially Mapped Crossover).

belong to the larger class of Evolutionary Algorithms (EAs), which are generic population-based metaheuristic optimization algorithms. All EAs and thus also GAs use biological evolution mechanisms such as representation, selection, recombination, and mutation. In terms of TSP and VRP the representation of the solution comprises encoding most important features of the solution as genes in a chromosome, which identifies the so-called individual in the population [25]. The recombination is carried out on two selected parent solutions by combining genes of parent chromosomes to create offspring solutions with potentially better quality. In turn, mutation is performed on the offspring solutions by random modification of genes to further explore the solution space and ensure genetic diversity. The GA is based on creating subsequent generations, and each new generation is constructed by selection, recombination, and mutation of all the solutions in the population. A properly designed GA should maintain a right balance between solution quality and diversity within the population to support efficient search. A common problem in the GAs for the routing problems is the feasibility of the solution created after recombination and mutation phases. An important issue is also the way how the offspring solutions are created, which led to designs of different crossover operators. Due to high efficiency of the GAs for solving various optimization problems, they are probably the most explored group of metaheuristics. Some of the most important crossover operators designed for the rich vehicle routing problems are shown in Table 4.1 [150]. Various genetic and evolutionary strategies and methods are described in the remaining part of this section. An illustration of the PMX crossover is shown in Fig. 4.18.

The edge assembly crossover operator (EAX) originally designed in 1997 for the TSP by Nagata and Kobayashi [125] was later extended to handle the VRP in 2007 by Nagata [122]. This powerful operator is based on specific se-

TABLE 4.1 Most important crossover operators.

Shortcut	Crossover name
EAX	edge assembly crossover
NX	natural crossover
ER	edge recombination crossover
MBX	matrix-based crossover
OX1	order crossover
OX2	order-based crossover
SMX	sorted match crossover
PMX	partially mapped crossover
MPX	maximal preservative crossover
MX1, MX2	merge crossover operators
CX	cycle crossover
POS	position-based crossover
AP	alternating position crossover
RC	route crossover
IB_X	insertion-based crossover
CPX	cluster preserving crossover
CEPX	common edges preserving crossover
BCRC	best cost route crossover

quence of steps combining edges from two parent solutions and in case of the VRP is described in the following way. In the first stage a graph is constructed containing edges from two parent solutions, followed by constructing cycles generated by alternately selecting a new edge from the first and second parents. Next, the subset of cycles is selected, and an intermediate solution is constructed by taking one parent and removing all edges available in the subset of cycles and by adding all edges in the subset of cycles from the second parent. Such an intermediate solution is then a combination of set of routes connected to the depot and subtours not connected to the depot. Next, a complete solution is created by repeatedly merging a subtour to a route with least cost strategy combined with a greedy heuristic. In the last stage the constraint violations are eliminated by applying penalty function, 2-Opt exchanges, and customer exchanges until solution becomes feasible [122]. An illustration of the EAX crossover is shown in Fig. 4.19.

The natural crossover was originally developed for the TSP in 2000 by Jung and Moon [88] and further extended to the VRPTW in 2002 [89]. The main idea of this crossover is to partition the set of customers from two parent solutions into two classes by drawing curves or geometric figures on graphical representation of the problem. As a result, the customers are located either on one side or the other of a curve, or the customers are either outside or inside the

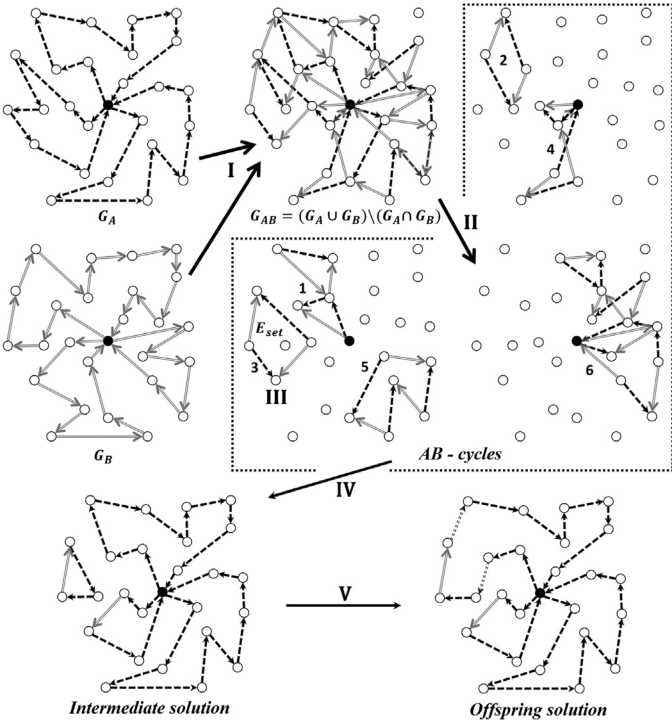

FIGURE 4.19 Illustration of the EAX (Edge Assembly Crossover).

geometric figures, such as ellipses, circles, or squares. The initial offspring solutions are created by transferring arcs from the first/second solution with both endings in the first/second class. They are further repaired by iterative attempts to connect disconnected segments from both parent solutions into the offsprings using the least cost strategy. In case a feasible solution is obtained, it is further improved by local optimization procedures based on 2-Opt, interroute, and intraroute search moves.

The Edge Recombination Crossover (ER) was originally developed for the TSP, and its main idea is to progressively extend tours by adding edges either from the first or second parent. Initially, all the edges from both parents are kept in a special edge table, in which all parental edges neighboring to a customer are grouped together. The ER was extended to the VRP in 1995 by Krajcar et al. [96] by using depot as a starting point of each route, by including the depot in the edge table, by considering only edges linked to the closest customers, and by considering capacity constraints to decide when to finish current route and start a new one.

The matrix-based crossover was originally developed for the TSP in 1991 by Fox and McMahon [53] and further extended in 1999 for the VRPTW by

Chin et al. [34]. In this crossover a matrix of size of customer numbers is used to indicate larger values in case customers a and b are in the same routes in two parent solutions and moreover they have the same precedence relationship. The values in the matrix for a given customer pair are smaller in case customers are not in the same route and intermediate when customers are in same routes but do not have the same precedence relationship. In the matrix-based crossover approach the offspring solutions are generated by iteratively inserting unserved customers into the routes with least insertion cost modified by the values in the matrix by checking all feasible insertion places.

The Order Crossover (OX1) was proposed in 1985 by Davis [40] and is used to build offspring solutions by selecting a subsequence of route from one parent and preserving the precedence order of customers from the other parent. The OX1 is used to copy the subtour between the crossover points directly the offspring solution, placing customers in same absolute position.

The Order-Based Crossover (OX2) was proposed in 1991 by Syswerda [175] and works by randomly selecting several customer positions in one parent route, and the corresponding order of customers is imposed on the second parent to create offspring solutions.

The Sorted Match Crossover (SMX) developed in 1985 by Brady [107] works by identifying subtours in both parent routes having same length, starting and ending at same customers, and containing the same customers. In case such subtours can be identified, the offspring solutions are created by replacing such a subtour with one having lower cost.

The Partially Mapped Crossover (PMX) was introduced in 1985 by Goldberg and Lingle [70] for the TSP, and it is based on selecting two random cut points on two parent solutions followed by creating offspring solutions by exchanging these subsequences between two parents. It is important to note that the order of remaining customers is tried to be preserved in as many customer cases as possible.

The Maximal Preservative Crossover (MPX) introduced in 1988 by Muhlenbein et al. [121] works similarly to the PMX by imposing additionally random subtour length to contain at least ten customers and to be smaller than or equal to the problem size divided by two. Having a subtour selected, all customers from it are removed from the second parent, and the offspring is generated by completing the random subtour with unserved customers in the same order as is in the second parent solution.

The Merge Crossover operators MX1 and MX2 for the VRPTW were proposed in 1993 by Blanton and Wainwright [19], and they were used to exploit a global precedence relationship between customers. Their main advantage especially for the VRPTW or PDPTW is that they rely on the assumption that is beneficiary for customer a to appear before customer b in a route in case time window of customer a starts earlier than time window of customer b. In MX1 and MX2 operators, two parent routes are checked position by position, selecting at each position customer with earlier start time window, who is later

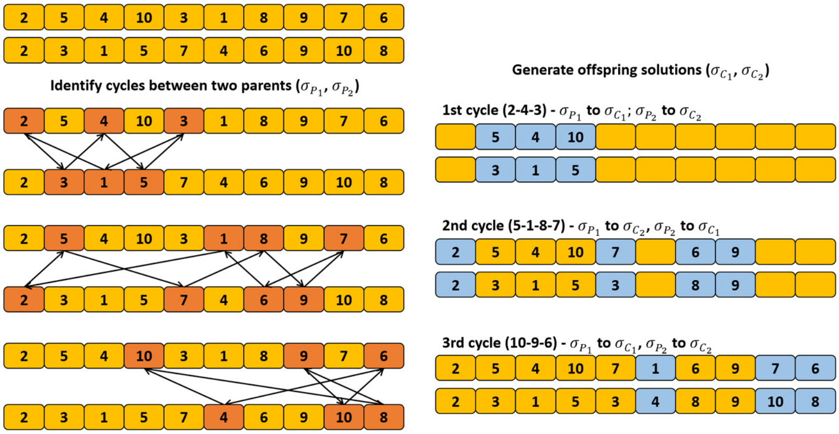

FIGURE 4.20 Illustration of the CX (Cycle Crossover).

transferred to the constructed offspring solution. By such a technique the generated offspring solution is biased toward solution containing customers sorted according to ascending order of their start time windows.

The Cycle Crossover (CX) was originally introduced in 1987 by Oliver et al. [131] for the TSP, and it is based on identifying cycles between two parent solutions using the connections between customer numbers and their positions in the routes. The customers that are included in the cycles will remain, whereas the other customers will be swapped to create offspring solutions. An illustration of the Cycle crossover is shown in Fig. 4.20.

The Position Based Crossover (POS) was developed in 1991 by Syswerda [175] and works similarly to the OX2 operator by randomly selecting customer positions in parent routes, but it imposes the position of selected customers on the corresponding customers in the second parent solution.

The Alternating Position Crossover (AP) was introduced in 1996 by Larranaga, and it creates offspring solutions by alternately selecting customers from the first and select parents, omitting customers already available in constructed offspring.

The Route Crossover (RC) was introduced in 1999 by Maeda et al. [108] based on the idea of bit masks corresponding to the number of routes in the first parent solution. The offspring solution is initially generated by selecting routes from first parent depending on bit mask values. All unserved customers are inserted into a temporary list, which is further sorted according to their order in the second part and then transferred one by one into generated offspring solution.

The Insertion-Based Crossover (IB_X) was proposed in 1998 by Berger et al. [17] based on iterative building and ruining solution taking into account large distances to successive customers and large waiting times. Initially, the first route is randomly selected from the first parent solution preferring routes containing customers with largest waiting times. Next, some customers are removed from such a route taking into account their large travel distance to successive customers or large waiting times to be served. In the next step a subset of routes from the second parent is selected that are close to this first route, and all such customers are further considered to be inserted into best positions in the first route of the first parent. Such a method is repeated for all routes from the first parent, resulting in an offspring solution. The second offspring solution is created by interchanging role of the parent from the first to the second one.

The Cluster-Preserving Crossover (CPX) and the Common Edges-Preserving Crossover (CEPX) operators were proposed in 2004 by Kubiak [97]. The CPX is used to preserve clusters or groups of customers that are common to both parent solutions. The routes having the largest number of customers are intersecting next to form clusters in the offspring solutions. Such clusters are used further to create offspring solutions by sequencing customers from clusters. The CEPX works similarly to CPX, but instead of clusters, it is used to preserve the common edges. Moreover, the longest subtours that are common to both parent solutions are used to generate routes in the offspring solutions.

The Best Cost Route Crossover (BCRC) was developed in 2006 by Ombuki et al. [133] for the VRP with Time Windows. In the first stage, two routes are randomly chosen from two parent solutions, and the customers from first route are removed from the second route, and in a similar way, customers from the second route are removed from the first route. The removed customers are reinserted back to original routes using the least-cost insertion methods.

Most of the genetic algorithms use one or more crossover operators combined with other methods such as guided local search, tabu search, simulated annealing, and other metaheuristics to provide best results for a given problem. The first genetic algorithms for the Vehicle Routing Problems were proposed in the 1990s, and since then, we may observe rapid development of the GAs until nowadays.

In 1991, Thangiah et al. [186] developed a genetic algorithm heuristic to solve the VRPTW, called GIDEON. This algorithm is a cluster-first route-second technique assigning customers to vehicles by the so-called Genetic Sectoring improved further by local optimization methods. The crossover used in GIDEON divides chromosomes at random points and exchanges subtours with other divided chromosomes.

Thangiah et al. [187] extended further this approach in 1994 by developing a hybrid genetic algorithm for the VRPTW. The initial solutions were created by Genetic Sectoring techniques and improved further by simulated annealing and tabu search methods.

In 1996, Potvin et al. [151] developed a new search technique for the VRPTW called the GENEtic ROUting System (GENEROUS) based on the paradigm of natural evolution. In this approach a population of solutions was evolving from one generation to another by merging two solutions into a single one, which is likely to be feasible with respect to the time window and capacity constraints.

Berger et al. [17] proposed in 1998 a new hybrid genetic algorithm for the VRPTW taking into account impact of explicit domain knowledge about and a priori characteristics of expected solutions used during the recombination and mutation phases of this algorithm. Their conceptually simple and easy-to-implement algorithm was designed to support time-constrained tasks and allowing for fast computation of near-optimal solutions, and thus it was used further in many other hybrid algorithms as the base model.

In 1999, Jih and Hsu [85] proposed a novel approach to solve single-vehicle Pickup and Delivery Problem with Time Windows by hybrid genetic algorithm. They incorporated usage of four crossover operators in the recombination phase of the GA: order crossover (OX1), order-based crossover (OX2), and merge crossover operators MX1 and MX2. Additionally, they included probability of mutation of a solution by three mutation schemes. The first mutation was based on randomly selecting two customers followed by interchanging their positions. The idea of second mutation was to randomly choose two cut customers and inverting the order of selected subroute. The third mutation was in turn based on checking if a vehicle arriving at the customer violated any of the constraints and then the order of customers in this subroute was randomly disturbed.

In 2003, Berger et al. [16] proposed a genetic algorithm for the VRPTW based on concurrently evolving two distinct populations of solutions, where the first population aimed the total travel distance, and the objective of the second population was to minimize the violations of the time window constraints. In the same year, Baker and Ayechew [9] compared a pure GA for the VRP with a hybrid GA combined with neighborhood search methods, proving that the hybrid method outperforms pure GA as well as tabu search and simulated annealing techniques.

Ho et al. [77] developed in 2008 two hybrid genetic algorithms for the Multi-Depot VRP, including in the first HGA random generating initial solutions, whereas the Clarke–Wright saving methods and the nearest neighbor heuristics were incorporated into the second HGA for the initialization phase. The results proved that simple generating random solutions in the first HGA was superior to the second considered technique.

In 2010, Ghoseiri and Ghannadpour [64] proposed a new solution for the Multi-Objective VRP with Time Windows by an interesting combination of goal programming and genetic algorithm in which the decision maker specifies optimistic aspiration levels to the objectives and deviations. In their GA, various heuristics incorporated local exploitation in the evolutionary search and the concept of Pareto optimality.

More recently, in 2015, Wang et al. [197] proposed a fitness-scaling adaptive GA with local search in which the fitness-scaling technique converts the raw fitness value to a new value suitable for further selection. Additionally, adaptive rates strategy was used to change the crossover and mutation probabilities depending on the fitness value, and a local search mechanism was applied to further exploit the problem space.

In 2017, Belhaiza et al. [13] developed a novel hybrid genetic variable neighborhood search algorithm for VRP with Multiple Time Windows. This algorithm encompasses combination of genetic crossover operators applied on list of best parent solutions with new implementations of local search operators.

Kumar and Panneerselvam [98] presented in 2017 a study of crossover operators for genetic algorithms to solve different variants of the VRP, and they also introduced a new Sinusoidal Motion Crossover (SMC) operator. This operator works analogously to sinusoidal motion of waves by alternately selecting consecutive customers from the first and second parents producing at the end two offspring solutions.

Baniamerian et al. [11] just recently in 2018 developed a hybrid metaheuristic combining the genetic algorithm hybridized with a modified variable neighborhood search to solve the vehicle routing problems with cross-docking. They suggested usage of some new four shaking and two neighborhood structures in a modified version of the VNS to solve the problem more efficiently in their combination with the GA.

4.2.17 Memetic Algorithms

The Memetic Algorithms (MAs) are population-based hybrid genetic algorithms hybridized with local search refinement procedures. They were introduced in 1989 by Moscato [118], and they are also commonly known as Genetic Local Search or Hybrid Evolutionary Algorithms because from optimization point of view, the evolutionary algorithm is used to perform exploration, whereas the local search methods are used to perform exploitation of the solution space. The MAs are inspired by models of adaptation in natural systems combining evolutionary adaptation of populations of individuals with individual learning within a lifetime [123]. The word *memetic* has its origin in word *meme* introduced in 1976 by Dawkins [41] to indicate the unit of imitation in cultural transmission and thus encompassing other forms of population-based techniques for optimization coming from a source different from genetic algorithms [120]. An illustration of the Memetic Algorithm is shown in Fig. 4.21.

The MAs are relatively a new group of specialized metaheuristics to solve various optimization problems, and initially they were considered to solve the TSP and simplest variants of the Vehicle Routing Problems [119]. In 1992, Moscato and Tinetti [120] developed a tree-structured memetic algorithm for the TSP, in which the optimizing population is divided into subpopulations and agents optimizing their current tours. Each agent is handling two tours, called *the pocket tour* and *current tour*, indicating best tour found so far and other tour being actually optimized by the heuristic assigned to the agent, respectively. All agents optimize their current tours with periods of local search, followed by recombination with other tours in case local minima is reached. The propagation of representative *memes* between subpopulations is taking place to spread out information about low-cost tours among all the agents.

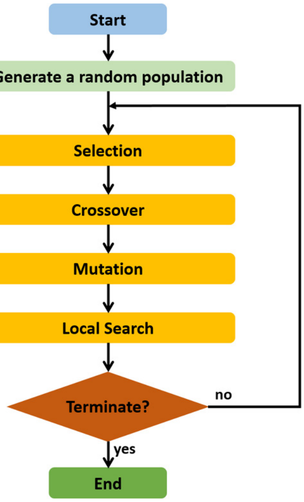

FIGURE 4.21 Illustration of the Memetic Algorithm.

In 1997, Nagata [125] proposed the Edge Assembly Crossover (EAX) operator for the TSP, which proved to be very powerful and was further extended and successfully used to the variants of the VRPs. This operator uses the edges from two parents to construct initial disjoint subtours, followed by connecting subtours in a greedy fashion using a construction similar to a minimal spanning tree to create offspring tours.

Merz and Freisleben [113] prepared in 2001 a good review on memetic algorithms for the TSP comparing the Maximum Preservative Crossover (MPX) by Gorges-Schleuter [71] and the Distance Preserving Crossover (DPX) by Freisleben and Merz [54]. The MPX generates offspring tours by copying a subtour between two randomly selected crossover points from the first parent and extending such partial tour by copying edges from the second parent. The DPX operator is very specific for the MAs as it is useful only if combined with local search [113]. It tries to generate offsprings by copying all items from the first parent and removing all edges not common between parents followed by reconnecting broken tour by local search methods. In 2002, Merz [112] further extended research on the MAs for the TSP by introducing the Generic Recombination Operator (GX), always preserving all common edges in offsprings. GX comprises four phases controlled by parameters reflecting the most important properties of recombination operators, and it proved to be superior to both MPX and DPX.

Labadi et al. [102] developed in 2008 a very efficient memetic algorithm for the VRP with Time Windows, which was able to minimize the total distance traveled also during the first phase of route minimization and improved 20 world's best-known solutions. The MA for the multicompartment VRP was proposed in 2008 by El Fallahi et al. [48], who combined memetic approach with a postoptimization phase based on path relinking and tabu search methods.

In 2009, Nagata and Braysy [123] developed a very powerful edge-assembly-based memetic algorithm for the Capacitated VRP. This algorithm combined the EAX crossover with well-known local search methods and additionally allowing for infeasible solutions due to capacity and route duration constraints after crossover operations. Their algorithm was able to find 20 new world's best-known solutions out of 47 standard benchmarks within very reasonable computational time. Nagata et al. [124] further extended the EAX crossover to be applied for the VRP with Time Windows in their penalty-based edge assembly memetic algorithm. They introduced adjustments of the EAX operator to the VRPTW and a penalty function to eliminate violations of both time window and capacity constraints from offspring solutions generated by the EAX operator. Their algorithm proved to be extremely powerful by finding 184 world's best-known solutions out of 356 benchmark instances.

Nalepa and Blocho [127] developed in 2016 an extension of the above method by introducing the adaptive version of the memetic algorithm for the VRPTW solving problem of automatic tuning of numerous hyperparameters. In their version the parameters of the algorithm, including the selection scheme, population size, and the number of child solutions, generated for each pair of parents, were adjusted dynamically during the search. Moreover, they introduced a new adaptive selection scheme to balance the exploration and exploitation of the solution space, which proved to be competitive and confirmed efficacy and convergence capabilities of the proposed approach.

A different approach was proposed in 2012 by Vidal et al. [190], who developed a very competitive hybrid genetic algorithm for solving the Capacitated VRP. They combined the exploration breadth of population-based evolutionary search with aggressive-improvement capabilities of neighborhood-based metaheuristics and advanced population-diversity management schemes, which allowed them to create competitive GA in terms of both computational efficiency and solution quality. They further extended these technique by introducing in 2013 [191] the Hybrid Genetic Search with Advanced Diversity Control for the VRPTW. This algorithm comprised new move evaluation techniques accounting for penalized infeasible solutions with respect to time window and duration constraints and allowing evaluation of moves from classical neighborhoods based on arc or node exchanges in amortized constant time. Moreover, they developed also geometric and structural problem decompositions to address large problems of the VRPTW efficiently.

The memetic algorithms were also taken into account while developing methods for the Pickup and Delivery Problems. In 2010, Nagata and Kobayashi [126] developed the MA for the PDP with Time Windows using Selective Route Exchange Crossover (SREX). Their algorithm allowed them to improve 146 best known solutions out of 298 instances. More recently, in 2017, Blocho and Nalepa [21] proposed a modification to the SREX based on the Longest

Common Subsequence (LCS) method combined with a new technique for representation of a solution to handle the crossover efficiently. Their approach was applied to the memetic algorithm for the PDPTW, proving ability to obtain high-quality feasible solutions.

4.3 Hyperheuristics for rich vehicle routing problems

The hyperheuristic is a new optimization paradigm, commonly described as heuristic to choose heuristics. It comprises search methods, learning techniques for generating or selecting heuristics to solve optimization problems [30]. The main difference between metaheuristics and hyperheuristics is that metaheuristics directly search a space of a problem, whereas hyperheuristics search a space of heuristics [116]. Heuristic selection and heuristic generation are two main classes of hyperheuristics approaches. The heuristic selection approach selecting and applying the most suitable heuristics from a set of problem-specific low-level heuristics (LLHs) at each problem solving state. On the other hand, heuristics generation techniques are used to automatically construct new heuristics from components of already existing heuristics, adjusted for the specific problem [29].

The heuristics selection techniques can be further divided into constructive and local-search hyperheuristics depending on the type of used LLHs. Similarly to construction heuristics (described in Section 4.1.1), the constructive hyperheuristics start with an empty solution building gradually a complete solution by subsequently choosing a proper low-level construction heuristic (selected from a set of LLHs) and by using it to enhance the quality of built solution. In turn, local-search hyperheuristics, similarly to improvement heuristics (described in Section 4.1.2), start from already generated solution and then try to gradually improve the quality of the solution by applying local searches and proper neighborhood structure.

The hyperheuristics usually can provide much better solutions than classical heuristics or metaheuristics applied to the specific problem individually. This is due that hyperheuristics can discover a good combination of best characteristics of individual low-level heuristics [116]. The crucial idea in hyperheuristics is solving optimization problems using various simple and flexible LLHs and developing a framework used to control the selection ad application of LLHs [60]. Typically, high-level heuristics and metaheuristics (variable neighborhood search, simulated annealing, tabu search, evolutionary algorithms, and many others described in Section 4.2) are used as search methods across the search space of heuristics.

In 2009, Garrido and Castro [60] proposed the Hill-climbing-based hyperheuristic for the CVRP problem. This technique works with ordered sequence of the so-called structures, and each structure defines one constructive and one improvement heuristic components. Initially, a random sequence of structures is generated with equal probability of each constructive and improvement heuristic. This sequence is next iteratively enhanced in a hill-climbing way trying to find a new neighbor sequence having better fitness than actual one. They also proposed some perturbing moves to add, delete, and replace structures and reallocating moves for exchanging customers between two adjacent structures.

Another hyperheuristic for the CVRP based on evolutionary algorithm was proposed by Garrido et al. [61] in 2009. This technique works with a population of sequences of structures, but contrary to the hill-climbing method described previously, structures may contain only one heuristic component and only of the same type, either constructive or improvement heuristics. This evolutionary hyperheuristic uses one recombination operator and eight mutation operators to generate new individuals. The population of individuals is evolving using a steady-state evolutionary model, where in each generation, one or two offsprings are constructed, which are further inserted into the main population.

In 2013, Mlejnek and Kubalík [116] proposed an evolutionary-based constructive heuristic selection hyperheuristic for the CVRP (HyperPOEMS), based on iterative local search algorithm. The HyperPOEMS operates on an ordered sequence of units containing selected constructive improvement heuristics. The design of that method allows for autonomous searching a structured space of various low-level heuristics to find proper combinations producing good solutions to the problem. The LLHs used in HyperPOEMS are various constructive (saving concept, sweep mechanism, cluster-first route-second, route-first cluster-second methods), improvement (2-opt, 3-opt, Or-opt, Van Breedam's moves) and order heuristics (increasing/decreasing demand, increasing/decreasing distance to the depot, radial sweeps). HyperPOEMS technique outperformed both hyperheuristics proposed by Garrido et al. [61].

Addressing limited availability of the constructive heuristics used by both Garrido et al. and Mlejnek and Kubalík, Drake et al. [46] proposed in 2013 a hyperheuristics methodology using Grammatical Evolution to simultaneously evolve constructive and perturbing heuristics. In this method a single initial solution is created by the construction heuristics, which is next iteratively enhanced by perturbation heuristics.

In 2016, Sim and Hart proposed a novel method for generating new constructive heuristics in a combined generative and selective hyperheuristic for the VRP. They tried to address the lack of available constructive methods and the potentially limiting quality of creating a weak candidate solution. This new construction heuristics may be used with any hyperheuristic methods requiring the creation of candidate solutions. Another important point is that they proposed also a new multipoint hyperheuristic method, called GP-MHH, comprising genetic programming in the first phase to evolve a population of construction heuristics, and a population of candidate solutions is constructed in the second phase by the evolved heuristics. The Memetic Hyperheuristic (MHH) is eventually applied on the population, which is a sophisticated perturbative-selection hyperheuristic. Based on extensive experimental studies, this method outperformed the grammatical evolution approach proposed by Drake et al. [46].

4.4 Summary

In this chapter, we described the most important heuristic methods for solving different variants of the Vehicle Routing Problems and Pickup and Delivery Problems. A heuristic approach to solve an optimization problem does not guarantee obtaining the optimal solution but enables to elaborate a feasible routing schedule in a very efficient way and in short time. Due to NP-hardness of the rich vehicle routing problems, solving problem instances to optimality is possible only in case of small-size tests. The main challenge in most of these problems is finding balance between available CPU time, size of the problem, and the quality of the approximated solution or need to find exact solution. Rapid development of various heuristic, metaheuristic, and hyperheuristic approaches helps in achieving both reduced computation time and better results.

In the first part of this chapter, we described the construction and improvement heuristics as two main groups of the classical heuristics. The construction heuristics build feasible solutions while trying to minimize its cost but often do not consider any improvement phases. The improvement heuristics are usually used for already generated solutions by other heuristics or exact algorithms. Local search methods are typically applied for simple local modifications, such as customer or arc exchanges, to generate neighboring solutions of possibly better quality.

The metaheuristics can be described as an enhancement of classical heuristics with emphasis on deep exploration of the solution space, and they usually combine sophisticated neighborhood search rules and recombination of solutions. The quality of the solutions produced by metaheuristics is typically much higher than that obtained by classical heuristics techniques but with a price of increased computing time. It is worth noting that the metaheuristic techniques are context dependent and usually require finely tuned parameters, which unfortunately make their extensions to other problems difficult.

Many various metaheuristics have been proposed for the vehicle routing problems, and they can be widely divided into local search, population search and learning mechanism groups; however, best metaheuristics merge ideas from different approaches. In the second part of this chapter, we described a wide range of the most popular metaheuristics algorithms.

The Simulated Annealing is a stochastic algorithm involving asymptotic convergence and allowing random movements in the searched neighborhood to escape local minima. Due to low-complexity, it can be used in many various optimization problems, not only related to the rich vehicle routing problems.

The Tabu Search is a search technique comprising local search methods and memory structures called tabu-list. Its main idea is to avoid cycling by inserting recently checked solution on the tabu-list, so during the search process, the solutions marked with tabu label are not taken into consideration. Such approach helps in getting out from the local minima and increases chances to find the global optimal solution.

The Adaptive Memory Procedure is based on the idea of the so-called adaptive memory being pool of good-quality solutions. The solutions in adaptive memory are always replaced with the new better-quality solutions coming from recombination of existing ones.

The Variable Neighborhood Search can be described as a framework for building heuristics, exploiting the neighborhoods both finding local optima and getting out of them by perturbation moves. In contrary to classical local search methods, the Variable Neighborhood Search is not based on following the trajectory but rather on exploring increasingly distant neighborhoods of a given solution and moving to the new one only in case of improvement. Such a method usually leads to maintaining best characteristics of current solution and helps obtain neighboring solutions of better quality.

The Large Neighborhood Search can be described as an iterative way of destroying and repairing the solution in the neighborhood. Destroying methods have some randomness such that different parts of the solution are destroyed and broader parts of the search tree are visited, and thus the searched neighborhood is larger than in classical local search methods.

The Greedy Randomized Adaptive Search Procedure is based on iterative two-phase search algorithm comprising construction and local search phases applied to various combinatorial optimization problems. In each iteration in the construction phase a feasible solution is constructed by a randomized greedy function. The solution is then iteratively improved in the second phase by local search movements.

The Particle Swarm Optimization aimed at producing computational intelligence by exploiting analogues of social interaction rather than individual cognitive abilities. It is an optimization method based on the idea of iterative improvements of a solution with regard to certain quality measures. The Particle Swarm Optimization is using a population of particles representing solutions to move them around in the search space due to some mathematical rules over particles velocity and position. Similarly to flock of birds collectively searching for food, the swarm is likely to move close to an optimum of the fitness function.

The Ant Colony Algorithm is based on the behavior of ants seeking paths between their colony and sources of food and laying some pheromone on trails. The fact that ants always follow the same path, which is indeed the shortest path, was the main motivation to take advantage of such real natural behavior of ants. The walking ants mark trails by laying down pheromones with information of quantity and quality of food. Such idea could be translated to the vehicle routing problems as searching in the neighborhood for good-quality solutions.

The Artificial Bee Colony algorithm is based on two fundamental concepts, namely self-organization and division of labor, which allow problem-solving systems to self-organize and adapt to the environment. The Artificial Bee Colony model related to collective intelligence of honey bees comprises employed and unemployed bees and food sources. It defines also recruitment to a nectar source and abandonment of such source as two leading methods of the honey bees behavior. The value of the food source for honey bees usually depends on its richness, proximity to the nest, and the ease of forage extracting, and they can be identified with good-quality solutions in the Artificial Bee Colony algorithm.

The Bat Algorithm is based on echolocation behavior of microbats when searching for their prey in nature.

The Cuckoo Search is a metaheuristic optimization algorithm inspired by the cuckoo birds being the "brood parasites" birds as they never build they own nests and instead lay their eggs in nests of other host bird nests. In terms of optimization algorithms, each egg represents a solution, and each cuckoo egg represents a new potentially better solution. The Cuckoo Search algorithm is based on the idea of creating subsequent generations of nests containing best eggs and thus highest quality solutions.

The Firefly Algorithm is inspired by flashing behavior of fireflies. The main purpose why fireflies flash is to signal and attract other flies.

The Golden Ball is a multipopulation metaheuristic based on soccer concept. The whole population of players (indicated as solutions) is created followed by random division of players into a fixed number of subpopulations called teams. Each team has its own couch related to training method, which is also randomly assigned to each time. The training method can be also associated with the way how each player evolves individually in the team, and in terms of the vehicle routing problems, it may be associated with local search moves.

The Gravitational Search Algorithm is based on the gravity law and mass interactions and was inspired by gravitation being the tendency of masses to accelerate toward each other. In the Gravitational Search Algorithm, agents are considered as objects, and their masses are related to their performance. All the agents attract each other by the gravity force communicating each other through gravitational force. The heavy objects, related to high-quality solutions, move slower than lighter objects, and this guarantees the exploitation step of this algorithm.

The idea of the Bacterial Foraging Optimization Algorithm is to mimic biological principles shown in the foraging behavior of E. coli bacteria for distributed optimization and control. In the Bacterial Foraging Optimization Algorithm a swarm of bacteria is used as searching agent for a solution to a specific optimization problem. A bacteria position represents a solution of the problem by a simple sequence of customer nodes. The bacteria direction vector is used to represent its ability to search for solution and is driving bacteria movement. In each step of the Bacterial Foraging Optimization Algorithm, all bacterium moves from one position to another are based on these directions, and by moving there are evaluated various solutions of the problem.

The Genetic Algorithms are optimization search methods based on the evolution process in nature, and they imitate the biological process of natural selection where stronger populations among different species survive. The Genetic Algorithms belong to the larger class of Evolutionary Algorithms, which are generic population-based metaheuristic optimization algorithms. All Evolutionary Algorithms and thus also Genetic Algorithms use biological evolution mechanisms such as representation, selection, recombination, and mutation. In terms of the rich vehicle routing problems, the representation of the solution comprises encoding most important features of the solution as genes in a chromosome, which identifies the so-called individual in the population. The recombination is carried out on two selected parent solutions by combining genes of parent chromosomes to create offspring solutions with potentially better quality. In turn, mutation is performed on the offspring solutions by random modification of genes to further explore the solution space and ensure genetic diversity. The Genetic Algorithm is based on creating subsequent generations, and each new generation is constructed by selection, recombination, and mutation of all the solutions in the population.

The Memetic Algorithms are population-based hybrid genetic algorithms hybridized with local search refinement procedures. From optimization point of view, the evolutionary algorithm is used to perform exploration, whereas the local search methods are used to perform exploitation. The MAs are inspired by models of adaptation in natural systems combining evolutionary adaptation of populations of individuals with individual learning within a lifetime.

The hyperheuristics is a new optimization paradigm, commonly described as heuristics to choose heuristics, comprising search methods, and learning techniques for generating or selecting heuristics to solve optimization problems. The main difference between metaheuristics and hyperheuristics is that metaheuristics directly search the solution space of a problem, whereas hyperheuristics search the space of heuristics.

References

[1] Emile Aarts, Jan Korst, Wil Michiels, Simulated Annealing, Springer US, Boston, MA, 2005, pp. 187–210.

[2] Mohamed M.S. Abdulkader, Yuvraj Gajpal, Tarek Y. ElMekkawy, Hybridized ant colony algorithm for the multi compartment vehicle routing problem, Applied Soft Computing 37 (2015) 196–203.

[3] Ala'a Abu-Srhan, E.A. Daoud, A hybrid algorithm using a genetic algorithm and cuckoo search algorithm to solve the traveling salesman problem and its application to multiple sequence alignment, 2014.

[4] Sohaib Afifi, Duc-Cuong Dang, Aziz Moukrim, A simulated annealing algorithm for the vehicle routing problem with time windows and synchronization constraints, in: Giuseppe Nicosia, Panos Pardalos (Eds.), Learning and Intelligent Optimization, Springer, Berlin, Heidelberg, 2013, pp. 259–265.

[5] The Jin Ai, Voratas Kachitvichyanukul, A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery, Computers & Operations Research 36 (5) (2009) 1693–1702, Selected papers presented at the Tenth International Symposium on Locational Decisions (ISOLDE X).

[6] Malek Alzaqebah, Salwani Abdullah, Sana Jawarneh, Modified artificial bee colony for the vehicle routing problems with time windows, SpringerPlus 5 (1) (Aug 2016) 1298.

[7] X. Yang, S. Deb, Cuckoo search via Lévy flights, in: 2009 World Congress on Nature Biologically Inspired Computing, NaBIC, Dec 2009, pp. 210–214.

[8] Nabila Azi, Michel Gendreau, Jean-Yves Potvin, An adaptive large neighborhood search for a vehicle routing problem with multiple routes, Computers & Operations Research 41 (2014) 167–173.

[9] Barrie M. Baker, M.A. Ayechew, A genetic algorithm for the vehicle routing problem, Computers & Operations Research 30 (5) (2003) 787–800.

[10] Edward K. Baker, Joanne R. Schaffer, Solution improvement heuristics for the vehicle routing and scheduling problem with time window constraints, American Journal of Mathematical and Management Sciences 6 (3–4) (1986) 261–300.

[11] Ali Baniamerian, Mahdi Bashiri, Fahime Zabihi, A modified variable neighborhood search hybridized with genetic algorithm for vehicle routing problems with cross-docking, Electronic Notes in Discrete Mathematics 66 (2018) 143–150, 5th International Conference on Variable Neighborhood Search.

[12] Benjamin Baran, Matilde Schaerer, A multiobjective ant colony system for vehicle routing problem with time windows 21 (01 2003) 97–102.

[13] S. Belhaiza, R. M'Hallah, G.B. Brahim, A new hybrid genetic variable neighborhood search heuristic for the vehicle routing problem with multiple time windows, in: 2017 IEEE Congress on Evolutionary Computation, CEC, June 2017, pp. 1319–1326.

[14] John E. Bell, Patrick R. McMullen, Ant colony optimization techniques for the vehicle routing problem, Advanced Engineering Informatics 18 (1) (2004) 41–48.

[15] Russell Bent, Pascal Van Hentenryck, A two-stage hybrid local search for the vehicle routing problem with time windows, Transportation Science 38 (4) (November 2004) 515–530.

[16] Jean Berger, Mohamed Barkaoui, A route-directed hybrid genetic approach for the vehicle routing problem with time windows, INFOR 41 (2003) 179–194.

[17] Jean Berger, Martin Salois, Regent Begin, A hybrid genetic algorithm for the vehicle routing problem with time windows, in: Robert E. Mercer, Eric Neufeld (Eds.), Advances in Artificial Intelligence, Springer, Berlin, Heidelberg, 1998, pp. 114–127.

[18] Ashita S. Bhagade, Vinayak Puranik, Artificial bee colony (ABC) algorithm for vehicle routing optimization problem.

[19] Joe L. Blanton Jr., Roger L. Wainwright, Multiple vehicle routing with time and capacity constraints using genetic algorithms, in: Proceedings of the 5th International Conference on Genetic Algorithms, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993, pp. 452–459.

[20] Andrius Blazinskas, Combining 2-opt, 3-opt and 4-opt with k-swap-kick perturbations for the traveling salesman problem, 2011.

[21] Miroslaw Blocho, Jakub Nalepa, LCS-based selective route exchange crossover for the pickup and delivery problem with time windows, in: Bin Hu, Manuel López-Ibáñez (Eds.), EvoCOP, in: Lecture Notes in Computer Science, vol. 10197, 2017, pp. 124–140.

[22] Burcin Bozkaya, Erhan Erkut, Gilbert Laporte, A tabu search heuristic and adaptive memory procedure for political districting, European Journal of Operational Research 144 (2003) 12–26.

[23] José Brandão, A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem, Computers & Operations Research 38 (2011) 140–151.

[24] Bräysy Olli, A reactive variable neighborhood search for the vehicle-routing problem with time windows, INFORMS Journal on Computing 15 (4) (2003) 347–368.

[25] Olli Bräysy, Wout Dullaert, Michel Gendreau, Evolutionary algorithms for the vehicle routing problem with time windows, Journal of Heuristics 10 (6) (2004) 587–611.

[26] Bräysy Olli, Michel Gendreau, Tabu search heuristics for the vehicle routing problem with time windows, Top 10 (2) (Dec 2002) 211–237.

[27] Olli Bräysy, Michel Gendreau, Vehicle routing problem with time windows, part I: route construction and local search algorithms, Transportation Science 39 (1) (February 2005) 104–118.

[28] Bernd Bullnheimer, Richard F. Hartl, Christine Strauss, Applying the ANT System to the Vehicle Routing Problem, Springer US, Boston, MA, 1999, pp. 285–296.

[29] Edmund K. Burke, Mathew R. Hyde, Graham Kendall, Gabriela Ochoa, Ender Ozcan, John R. Woodward, Exploring Hyper-Heuristic Methodologies With Genetic Programming, Springer, Berlin, Heidelberg, 2009, pp. 177–201.

[30] Edmund K. Burke, Matthew Hyde, Graham Kendall, Gabriela Ochoa, Ender Özcan, John R. Woodward, A Classification of Hyper-Heuristic Approaches, Springer US, Boston, MA, 2010, pp. 449–468.

[31] Ann Melissa Campbell, Martin Savelsbergh, Efficient insertion heuristics for vehicle routing and scheduling problems, Transportation Science 38 (3) (August 2004) 369–378.

[32] Huiqin Chen, Sheng Li, Zheng Tang, Hybrid gravitational search algorithm with random-key encoding scheme combined with simulated annealing, International Journal of Computer Science and Network Security 11 (01 2011).

[33] Xichun Chen, Junli Wang, A novel hybrid cuckoo search algorithm for optimizing vehicle routing problem in logistics distribution system, Journal of Computational and Theoretical Nanoscience 13 (1) (2016).

[34] A. Chin, H. Kit, A. Lim, A new GA approach for the vehicle routing problem, in: Proceedings 11th International Conference on Tools With Artificial Intelligence, Nov 1999, pp. 307–310.

[35] G. Clarke, J.W. Wright, Scheduling of vehicles from a central depot to a number of delivery points, Operations Research 12 (4) (August 1964) 568–581.

[36] Alberto Colorni, Marco Dorigo, Vittorio Maniezzo, Distributed optimization by ant colonies, 1991.

[37] Jean-Francois Cordeau, Gilbert Laporte, Tabu Search Heuristics for the Vehicle Routing Problem, Springer US, Boston, MA, 2005, pp. 145–163.

[38] Jean-Francois Cordeau, Martin W.P. Savelsbergh, Chapter 6: Vehicle routing, 2006.

[39] A. Croes, A method for solving traveling salesman problems, Operations Research 5 (1958) 791–812.

[40] Lawrence Davis, Applying adaptive algorithms to epistatic domains, in: Proceedings of the 9th International Joint Conference on Artificial Intelligence – vol. 1, IJCAI'85, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1985, pp. 162–164.

[41] R. Dawkins, The Selfish Gene, Oxford University Press, Oxford, UK, 1976.

[42] Kenneth Alan De Jong, An Analysis of the Behavior of a Class of Genetic Adaptive Systems, PhD thesis, Ann Arbor, MI, USA, AAI7609381, 1975.

[43] Alberto V. Donati, Roberto Montemanni, Norman Casagrande, Andrea E. Rizzoli, Luca M. Gambardella, Time dependent vehicle routing problem with a multi ant colony system, European Journal of Operational Research 185 (3) (2008) 1174–1191.

[44] Marco Dorigo, Optimization, Learning and Natural Algorithms, PhD thesis, Politecnico di Milano, Italy, 1992.

[45] Mohammad Bagher Dowlatshahi, Hossein Nezamabadi-pour, Mashaallah Mashinchi, A discrete gravitational search algorithm for solving combinatorial optimization problems, Information Sciences 258 (2014) 94–107.

[46] John H. Drake, Nikolaos Kililis, Ender Özcan, Generation of VNS components with grammatical evolution for vehicle routing, in: Krzysztof Krawiec, Alberto Moraglio, Ting Hu, A. Şima Etaner-Uyar, Bin Hu (Eds.), Genetic Programming, Springer, Berlin, Heidelberg, 2013, pp. 25–36.

[47] Christophe Duhamel, Philippe Lacomme, Christian Prins, Caroline Prodhon, A GRASPxELS approach for the capacitated location-routing problem, Computers & Operations Research 37 (11) (2010) 1912–1923, Metaheuristics for Logistics and Vehicle Routing.

[48] Abdellah El-Fallahi, Christian Prins, Roberto Wolfler Calvo, A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem, Computers & Operations Research 35 (5) (2008) 1725–1741.

[49] Özlem Ergun, James B. Orlin, Abran Steele-Feldman, Creating very large scale neighborhoods out of smaller ones by compounding moves, Journal of Heuristics 12 (1) (Mar 2006) 115–140.

[50] Ana Laura Vázquez-Esquivel, Jesús Aguirre-García, Andrea Isabel Cervantes-Payan, Edgar Osvaldo Escobedo-Hernández, Ernesto Liñán-García, Helue I. De la Barrera-Gómez, Luis A. López-Alday, Solving vehicle routing problem with multi-phases simulated annealing algorithm, 2018.
[51] Thomas A. Feo, Mauricio G.C. Resende, Greedy randomized adaptive search procedures, Journal of Global Optimization 6 (2) (Mar 1995) 109–133.
[52] Merrill M. Flood, The traveling-salesman problem, Operations Research 4 (1) (1956) 61–75.
[53] B.R. Fox, M.B. McMahon, Genetic operators for sequencing problems, 1990.
[54] Bernd Freisleben, Peter Merz, A genetic local search algorithm for solving symmetric and asymmetric traveling salesman problems, in: International Conference on Evolutionary Computation, 1996, pp. 616–621.
[55] Guenther Fuellerer, Karl F. Doerner, Richard F. Hartl, Manuel Iori, Ant colony optimization for the two-dimensional loading vehicle routing problem, Computers & Operations Research 36 (3) (2009) 655–673.
[56] Y. Gajpal, P. Abad, Saving-based algorithms for vehicle routing problem with simultaneous pickup and delivery, The Journal of the Operational Research Society 61 (10) (2010) 1498–1509.
[57] Yuvraj Gajpal, Prakash Abad, An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup, Computers & Operations Research 36 (12) (2009) 3215–3223, New developments on hub location.
[58] L.M. Gambardella, E. Taillard, G. Agazzi, MACS-VRPTW: a Multiple Ant Colony System for Vehicle Routing Problems With Time Windows, Technical report, 1999.
[59] Xiaobing Gan, Lijiao Liu, Ben Niu, L.J. Tan, F.F. Zhang, J. Liu, SRBFOs for solving the heterogeneous fixed fleet vehicle routing problem, in: De-Shuang Huang, Kang-Hyun Jo, Abir Hussain (Eds.), Intelligent Computing Theories and Methodologies, Springer International Publishing, Cham, 2015, pp. 725–732.
[60] Pablo Garrido, Carlos Castro, Stable solving of CVRPs using hyperheuristics, in: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, GECCO'09, ACM, New York, NY, USA, 2009, pp. 255–262.
[61] Pablo Garrido, Carlos Castro, Eric Monfroy, Towards a flexible and adaptable hyperheuristic approach for VRPs 1 (01 2009) 311–317.
[62] Michel Gendreau, Alain Hertz, Gilbert Laporte, New insertion and postoptimization procedures for the traveling salesman problem, Operations Research 40 (1086–1094) (12 1992).
[63] Michel Gendreau, Alain Hertz, Gilbert Laporte, A tabu search heuristic for the vehicle routing problem, Management Science 40 (10) (1994) 1276–1290.
[64] Keivan Ghoseiri, Seyed Farid Ghannadpour, Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm, Applied Soft Computing 10 (4) (2010) 1096–1107, Optimisation Methods & Applications in Decision-Making Processes.
[65] Fred Glover, Tabu search – part I, INFORMS Journal on Computing 1 (3) (1989) 190–206.
[66] Fred Glover, Ejection chains, reference structures and alternating path methods for traveling salesman problems, Discrete Applied Mathematics 65 (1) (1996) 223–253, First International Colloquium on Graphs and Optimization.
[67] Rajeev Goel, Raman Maini, A hybrid of ant colony and firefly algorithms (HAFA) for solving vehicle routing problems, Journal of Computational Science 25 (2018) 28–37.
[68] Fatma Pinar Goksal, Ismail Karaoglan, Fulya Altiparmak, A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery, Computers & Industrial Engineering 65 (1) (2013) 39–53, Intelligent Manufacturing Systems.
[69] David E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, 1st edition, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
[70] David E. Goldberg, Robert Lingle Jr., Alleles, loci, and the traveling salesman problem.
[71] Martina Gorges-Schleuter, Genetic Algorithms and Population Structures: a Massively Parallel Algorithm, PhD thesis, Uni Dortmund, 1991.

[72] Chrysanthos E. Gounaris, Panagiotis P. Repoussis, Christos D. Tarantilis, Wolfram Wiesemann, Christodoulos A. Floudas, An adaptive memory programming framework for the robust capacitated vehicle routing problem, Transportation Science 50 (4) (2016) 1239–1260.
[73] Philippe Grangier, Michel Gendreau, Fabien Lehuédé, Louis-Martin Rousseau, An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization, European Journal of Operational Research 254 (1) (2016) 80–91.
[74] Lahcene Guezouli, Mohamed Bensakhria, Samir Abdelhamid, Efficient golden-ball algorithm based clustering to solve the multi-depot VRP with time windows, International Journal of Applied Evolutionary Computation 9 (2018) 1–16.
[75] Vera C. Hemmelmayr, Jean-Francois Cordeau, Teodor Gabriel Crainic, An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics, Computers & Operations Research 39 (12) (2012) 3215–3228.
[76] Seda Hezer, Yakup Kara, Solving vehicle routing problem with simultaneous delivery and pick-up using an algorithm based on bacterial foraging optimization, Journal of the Faculty of Engineering and Architecture of Gazi University 28 (06 2013) 373–382.
[77] William Ho, George T.S. Ho, Ping Ji, Henry C.W. Lau, A hybrid genetic algorithm for the multi-depot vehicle routing problem, Engineering Applications of Artificial Intelligence 21 (4) (2008) 548–557.
[78] John H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control and Artificial Intelligence, MIT Press, Cambridge, MA, USA, 1992.
[79] Ali Asghar Rahmani Hosseinabadi, Maryam Kardgar, Mohammad Shojafar, Shahab Shamshirband, Ajith Abraham, Gravitational search algorithm to solve open vehicle routing problem, in: Václav Snášel, Ajit Abraham, Pavel Krömer, Millie Pant, Azah Kamilah Muda (Eds.), Innovations in Bio-Inspired Computing and Applications, Springer International Publishing, Cham, 2016, pp. 93–103.
[80] Aliasghar Rahmani Hosseinabadi, Mohammad Yazdanpanah, Ali Shokouhi Rostami, A new search algorithm for solving symmetric traveling salesman problem based on gravity, 2013.
[81] Wenbin Hu, Huanle Liang, Chao Peng, Bo Du, Qi Hu, A hybrid chaos-particle swarm optimization algorithm for the vehicle routing problem with time window, Entropy [electronic only] 4 (04 2013).
[82] G.K. Jati, H.M. Manurung, Suyanto, Discrete cuckoo search for traveling salesman problem, in: 2012 7th International Conference on Computing and Convergence Technology, ICCCT, Dec 2012, pp. 993–997.
[83] Gilang Kusuma Jati, Suyanto, Evolutionary discrete firefly algorithm for travelling salesman problem, in: Abdelhamid Bouchachia (Ed.), Adaptive and Intelligent Systems, Springer, Berlin, Heidelberg, 2011, pp. 393–403.
[84] Liu Jie, Lin Teng, Shoulin Yin, An improved discrete firefly algorithm used for traveling salesman problem, in: Ying Tan, Hideyuki Takagi, Yuhui Shi (Eds.), Advances in Swarm Intelligence, Springer International Publishing, Cham, 2017, pp. 593–600.
[85] Wan-Rong Jih, J. Yung-Jen Hsu, Dynamic vehicle routing using hybrid genetic algorithms, in: Proceedings 1999 IEEE International Conference on Robotics and Automation, Cat. No. 99CH36288C, vol. 1, May 1999, pp. 453–458.
[86] A.S. Joshi, Omkar Kulkarni, G.M. Kakandikar, V.M. Nandedkar, Cuckoo search optimization – a review, Materials Today: Proceedings 4 (8) (2017) 7262–7269, International Conference on Advancements in Aeromechanical Materials for Manufacturing (ICAAMM-2016): Organized by MLR Institute of Technology, Hyderabad, Telangana, India.
[87] J.W. Joubert, S.J. Claasen, A sequential insertion heuristic for the initial solution to a constrained vehicle routing problem, ORiON 22 (06 2006).
[88] Soonchul Jung, Byung-Ro Moon, The natural crossover for the 2D Euclidean TSP, in: Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation, GECCO'00, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000, pp. 1003–1010.

[89] Soonchul Jung, Byung-Ro Moon, A hybrid genetic algorithm for the vehicle routing problem with time windows, in: Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation, GECCO'02, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002, pp. 1309–1316.

[90] Can B. Kalayci, Can Kaya, An ant colony system empowered variable neighborhood search algorithm for the vehicle routing problem with simultaneous pickup and delivery, Expert Systems with Applications 66 (2016) 163–175.

[91] Karaboga Dervis, An Idea Based on Honey Bee Swarm for Numerical Optimization, Technical report – tr06, Erciyes University, 01 2005.

[92] James Kennedy, Russell Eberhart, Particle swarm optimization 4 (12 1995) 1942–1948.

[93] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.

[94] Gözde Kizilateş, Fidan Nuriyeva, On the nearest neighbor algorithms for the traveling salesman problem, in: Dhinaharan Nagamalai, Ashok Kumar, Annamalai Annamalai (Eds.), Advances in Computational Science, Engineering and Information Technology, Springer International Publishing, Heidelberg, 2013, pp. 111–118.

[95] George Kontoravdis, Jonathan Bard, A grasp for the vehicle routing problem with time windows, INFORMS Journal on Computing 7 (02 1995) 10–23.

[96] Slavko Krajcar, Davor Skrlec, Branko Pribicevic, Snjezana Blagajac, Ga approach to solving multiple vehicle routing problem, in: Carlos Pinto-Ferreira, Nuno J. Mamede (Eds.), Progress in Artificial Intelligence, Springer, Berlin, Heidelberg, 1995, pp. 473–481.

[97] Marek Kubiak, Systematic construction of recombination operators for the vehicle routing problem 29 (01 2004).

[98] S.G. Varun Kumar, Ramasamy Panneerselvam, A study of crossover operators for genetic algorithms to solve VRP and its variants and new sinusoidal motion crossover operator, International Journal of Computational Intelligence Research 13 (06 2017) 1717–1733.

[99] Sharad N. Kumbharana, Gopal M. Pandey, Solving travelling salesman problem using firefly algorithm, International Journal for Research in Science & Advanced Technologies (March–April 2013).

[100] R.J. Kuo, Ferani E. Zulvia, Kadarsah Suryadi, Hybrid particle swarm optimization with genetic algorithm for solving capacitated vehicle routing problem with fuzzy demand – a case study on garbage collection system, Applied Mathematics and Computation 219 (5) (2012) 2574–2588.

[101] Jari Kytöjoki, Teemu Nuortio, Olli Bräysy, Michel Gendreau, An efficient variable neighborhood search heuristic for very large scale vehicle routing problems, Computers & Operations Research 34 (9) (2007) 2743–2757.

[102] Nacima Labadi, Christian Prins, Mohamed Reghioui, A memetic algorithm for the vehicle routing problem with time windows, RAIRO - Operations Research 42 (3) (2008) 415–431.

[103] G. Laporte, F. Semet, Classical heuristics for the capacitated VRP, in: The Vehicle Routing Problem, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001, pp. 109–128.

[104] Bo Li, Chen Guo, Tao Ning, Yingqi Wei, Solve VRPPD with improved bacteria optimization algorithm, 01 2017.

[105] Xiangyong Li, Peng Tian, Y.P. Aneja, An adaptive memory programming metaheuristic for the heterogeneous fixed fleet vehicle routing problem, Transportation Research Part E: Logistics and Transportation Review 46 (6) (2010) 1111–1127.

[106] F-H. Liu, S-Y. Shen, The fleet size and mix vehicle routing problem with time windows, Journal of the Operational Research Society 50 (7) (Jul 1999) 721–732.

[107] R.M. Brady, Optimization strategies gleaned from biological evolution, Nature 317 (10 1985) 804–806.

[108] O. Maeda, M. Nakamura, B.M. Ombuki, K. Onaga, A genetic algorithm approach to vehicle routing problem with time deadlines in geographical information systems, in: IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics, Cat. No. 99CH37028, vol. 4, Oct 1999, pp. 595–600.

[109] Yannis Marinakis, Multiple phase neighborhood search-grasp for the capacitated vehicle routing problem, Expert Systems with Applications 39 (8) (2012) 6807–6815.
[110] Yannis Marinakis, Magdalene Marinaki, Georgios Dounias, A hybrid particle swarm optimization algorithm for the vehicle routing problem, Engineering Applications of Artificial Intelligence 23 (4) (2010) 463–472.
[111] Rafael Mart, Gerhard Reinelt, The Linear Ordering Problem: Exact and Heuristic Methods in Combinatorial Optimization, 1st edition, Springer Publishing Company, Incorporated, 2011.
[112] Peter Merz, A comparison of memetic recombination operators for the traveling salesman problem, in: William B. Langdon, Erick Cantú-Paz, Keith E. Mathias, Rajkumar Roy, David Davis, Riccardo Poli, Karthik Balakrishnan, Vasant G. Honavar, Günter Rudolph, Joachim Wegener, Larry Bull, Mitchell A. Potter, Alan C. Schultz, Julian F. Miller, Edmund K. Burke, Natasa Jonoska (Eds.), GECCO, Morgan Kaufmann, 2002, pp. 472–479.
[113] Peter Merz, Bernd Freisleben, Memetic algorithms for the traveling salesman problem, Complex Systems 13 (4) (2002).
[114] S.A. MirHassani, N. Abolghasemi, A particle swarm optimization algorithm for open vehicle routing problem, Expert Systems with Applications 38 (9) (2011) 11547–11551.
[115] Nenad Mladenovic, Pierre Hansen, Variable neighborhood search, Computers & Operations Research 24 (11) (1997) 1097–1100.
[116] Jaromír Mlejnek, Jiří Kubalik, Evolutionary hyperheuristic for capacitated vehicle routing problem, in: Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation, GECCO'13 Companion, ACM, New York, NY, USA, 2013, pp. 219–220.
[117] L. Moccia, J-F. Cordeau, G. Laporte, An incremental tabu search heuristic for the generalized vehicle routing problem with time windows, Journal of the Operational Research Society 63 (2) (February 2012) 232–244.
[118] Pablo Moscato, On evolution, search, optimization, genetic algorithms and martial arts – towards memetic algorithms, 1989.
[119] Pablo Moscato, Carlos Cotta, A Gentle Introduction to Memetic Algorithms, Springer US, Boston, MA, 2003, pp. 105–144.
[120] Pablo Moscato, Fernando Tinetti, Blending Heuristics With a Population-Based Approach: a "Memetic" Algorithm for the Traveling Salesman Problem, Report 92-12, Universidad Nacional de La Plata, C.C. 75, 1900 La Plata, 1994.
[121] H. Mühlenbein, M. Gorges-Schleuter, O. Krämer, Evolution algorithms in combinatorial optimization, Parallel Computing 7 (1) (1988) 65–85.
[122] Yuichi Nagata, Edge assembly crossover for the capacitated vehicle routing problem, in: Carlos Cotta, Jano van Hemert (Eds.), Evolutionary Computation in Combinatorial Optimization, Springer, Berlin, Heidelberg, 2007, pp. 142–153.
[123] Yuichi Nagata, Olli Bräysy, Edge assembly-based memetic algorithm for the capacitated vehicle routing problem, Networks 54 (4) (December 2009) 205–215.
[124] Yuichi Nagata, Olli Bräysy, Wout Dullaert, A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows, Computers & Operations Research 37 (4) (2010) 724–737.
[125] Yuichi Nagata, Shigenobu Kobayashi, Edge assembly crossover: a high-power genetic algorithm for the travelling salesman problem, in: ICGA, 1997.
[126] Yuichi Nagata, Shigenobu Kobayashi, A memetic algorithm for the pickup and delivery problem with time windows using selective route exchange crossover, in: Robert Schaefer, Carlos Cotta, Joanna Kolodziej, Günter Rudolph (Eds.), PPSN (1), in: Lecture Notes in Computer Science, vol. 6238, Springer, 2010, pp. 536–545.
[127] Jakub Nalepa, Miroslaw Blocho, Adaptive memetic algorithm for minimizing distance in the vehicle routing problem with time windows, Soft Computing 20 (6) (2016) 2309–2327.
[128] K.K.H. Ng, C.K.M. Lee, S.Z. Zhang, Kan Wu, William Ho, A multiple colonies artificial bee colony algorithm for a capacitated vehicle routing problem and re-routing strategies under time-dependent traffic congestion, Computers & Industrial Engineering 109 (2017) 151–168.

[129] Phuong Khanh Nguyen, Teodor Gabriel Crainic, Michel Toulouse, A tabu search for time-dependent multi-zone multi-trip vehicle routing problem with time windows, European Journal of Operational Research 231 (1) (2013) 43–56.

[130] Ben Niu, Hong Wang, Li-Jing Tan, Li Li, Jing-Wen Wang, Vehicle routing problem with time windows based on adaptive bacterial foraging optimization, in: De-Shuang Huang, Jianhua Ma, Kang-Hyun Jo, M. Michael Gromiha (Eds.), Intelligent Computing Theories and Applications, Springer, Berlin, Heidelberg, 2012, pp. 672–679.

[131] I.M. Oliver, D.J. Smith, J.R.C. Holland, A study of permutation crossover operators on the traveling salesman problem, in: ICGA, 1987.

[132] Alfredo Olivera, Omar Viera, Adaptive memory programming for the vehicle routing problem with multiple trips, Computers & Operations Research 34 (1) (2007) 28–47.

[133] Beatrice Ombuki, Brian J. Ross, Franklin Hanshar, Multi-objective genetic algorithms for vehicle routing problem with time windows, Applied Intelligence 24 (1) (Feb 2006) 17–30.

[134] İlhan Or, Traveling Salesman-Type Combinatorial Problems and Their Relation to the Logistics of Regional Blood Banking, Ph. D., NorthWestern University, Illinois, 1967.

[135] E. Osaba, F. Diaz, E. Onieva, Golden ball: a novel meta-heuristic to solve combinatorial optimization problems based on soccer concepts, Applied Intelligence 41 (1) (Jul 2014) 145–166.

[136] Eneko Osaba, Roberto Carballedo, Pedro Lopez-Garcia, Fernando Diaz, Comparison between golden ball meta-heuristic, evolutionary simulated annealing and tabu search for the traveling salesman problem, in: Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion, GECCO'16 Companion, ACM, New York, NY, USA, 2016, pp. 1469–1470.

[137] Eneko Osaba, Roberto Carballedo, Xin-She Yang, Fernando Diaz, An Evolutionary Discrete Firefly Algorithm With Novel Operators for Solving the Vehicle Routing Problem With Time Windows, Springer International Publishing, Cham, 2016, pp. 21–41.

[138] Eneko Osaba, Roberto Carballedo, Xin-She Yang, Iztok Fister Jr., Pedro Lopez-Garcia, Javier Del Ser, On Efficiently Solving the Vehicle Routing Problem With Time Windows Using the Bat Algorithm With Random Reinsertion Operators, Springer International Publishing, Cham, 2018, pp. 69–89.

[139] Eneko Osaba, X.S. Yang, Fernando Díaz, Enrique Onieva, Antonio D. Masegosa, Asier Perallos, A bio-inspired discrete firefly algorithm to solve a rich vehicle routing problem modeling a newspaper distribution problem with recycling policy, 2015.

[140] Ibrahim Hassan Osman, Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem, Annals of Operations Research 41 (4) (Dec 1993) 421–451.

[141] Aziz Ouaarab, Belaïd Ahiod, Xin-She Yang, Random-key cuckoo search for the travelling salesman problem, Soft Computing 19 (4) (Apr 2015) 1099–1106.

[142] Xinxin Ouyang, Yongquan Zhou, Qifang Luo, Huan Chen, A novel discrete cuckoo search algorithm for spherical traveling salesman problem, 2013.

[143] Fengshan Pan, Chunming Ye, Kefeng Wang, Jiangbo Cao, Research on the vehicle routing problem with time windows using firefly algorithm, Journal of Computers 8 (09 2013).

[144] D.C. Paraskevopoulos, P.P. Repoussis, C.D. Tarantilis, G. Ioannou, G.P. Prastacos, A reactive variable neighborhood tabu search for the heterogeneous fleet vehicle routing problem with time windows, Journal of Heuristics 14 (5) (Oct 2008) 425–455.

[145] K.M. Passino, Biomimicry of bacterial foraging for distributed optimization and control, IEEE Control Systems Magazine 22 (3) (June 2002) 52–67.

[146] Tantikorn Pichpibul, Ruengsak Kawtummachai, New enhancement for Clarke–Wright savings algorithm to optimize the capacitated vehicle routing problem, European Journal of Scientific Research 78 (06 2012).

[147] Telmo Pinto, C. Alves, José Carvalho, Ana Moura, An insertion heuristic for the capacitated vehicle routing problem with loading constraints and mixed linehauls and backhauls 43 (01 2015) 311–318.

[148] Michael Polacek, Richard F. Hartl, Karl Doerner, Marc Reimann, A variable neighborhood search for the multi depot vehicle routing problem with time windows, Journal of Heuristics 10 (6) (Dec 2004) 613–627.

[149] Riccardo Poli, James Kennedy, Tim Blackwell, Particle swarm optimization, Swarm Intelligence 1 (1) (Jun 2007) 33–57.

[150] Jean-Yves Potvin, Evolutionary algorithms for vehicle routing, 2007.

[151] Jean-Yves Potvin, Samy Bengio, The vehicle routing problem with time windows – part II: genetic search, INFORMS Journal on Computing 8 (2) (1996) 165–172.

[152] Jean-Yves Potvin, Tanguy Kervahut, Bruno-Laurent Garcia, Jean-Marc Rousseau, The vehicle routing problem with time windows – part I: tabu search, INFORMS Journal on Computing 8 (2) (1996) 158–164.

[153] Jean-Yves Potvin, Jean-Marc Rousseau, An exchange heuristic for routing problems with time windows, Journal of the Operational Research Society 46 (12) (Dec 1995) 1433–1446.

[154] Christian Prins, A GRASP × Evolutionary Local Search Hybrid for the Vehicle Routing Problem, Springer, Berlin, Heidelberg, 2009, pp. 35–53.

[155] Esmat Rashedi, Hossein Nezamabadi-pour, Saeid Saryazdi, GSA: a gravitational search algorithm, Information Sciences 179 (13) (2009) 2232–2248, Special Section on High Order Fuzzy Sets.

[156] Martin Reed, Aliki Yiannakou, Roxanne Evering, An ant colony algorithm for the multi-compartment vehicle routing problem, Applied Soft Computing 15 (2014) 169–176.

[157] P.P. Repoussis, C.D. Tarantilis, Solving the fleet size and mix vehicle routing problem with time windows via adaptive memory programming, Transportation Research Part C: Emerging Technologies 18 (5) (2010) 695–712, Applications of Advanced Technologies in Transportation: Selected papers from the 10th AATT Conference.

[158] Glaydston Mattos Ribeiro, Gilbert Laporte, An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem, Computers & Operations Research 39 (3) (2012) 728–735.

[159] Yves Rochat, Éric D. Taillard, Probabilistic diversification and intensification in local search for vehicle routing, Journal of Heuristics 1 (1) (Sep 1995) 147–167.

[160] Stefan Ropke, David Pisinger, An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows, Transportation Science 40 (4) (2006) 455–472.

[161] Robert A. Russell, An effective heuristic for the m-tour traveling salesman problem with some side conditions, Operations Research 25 (3) (1977) 517–524.

[162] K. Ruttanateerawichien, W. Kurutach, T. Pichpibul, A new efficient and effective golden-ball-based technique for the capacitated vehicle routing problem, in: 2016 IEEE/ACIS 15th International Conference on Computer and Information Science, ICIS, June 2016, pp. 1–5.

[163] Kanjana Ruttanateerawichien, Werasak Kurutach, Tantikorn Pichpibul, An improved golden ball algorithm for the capacitated vehicle routing problem, in: Linqiang Pan, Gheorghe Păun, Mario J. Pérez-Jiménez, Tao Song (Eds.), Bio-Inspired Computing – Theories and Applications, Springer, Berlin, Heidelberg, 2014, pp. 341–356.

[164] S. Salhi, G. Nagy, A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling, Journal of the Operational Research Society 50 (10) (Oct 1999) 1034–1042.

[165] Martin W.P. Savelsbergh, The vehicle routing problem with time windows: minimizing route duration, INFORMS Journal on Computing 4 (1992) 146–154.

[166] Fatima Sayoti, Mohammed Riffi, Random-keys golden ball algorithm for solving traveling salesman problem, International Review on Modelling and Simulations (IREMOS) 8 (02 2015).

[167] B. Schutz, Gravity From the Ground up, December 2003.

[168] Davoud Sedighizadeh, Houman Mazaheripour, Optimization of multi objective vehicle routing problem using a new hybrid algorithm based on particle swarm optimization and artificial bee colony algorithm considering precedence constraints, Alexandria Engineering Journal 57 (4) (2018) 2225–2239.

[169] Frédéric Semet, Éric D. Taillard, Solving real-life vehicle routing problems efficiently using tabu search, Annals of Operations Research 41 (4) (1993) 469–488.

[170] Paul Shaw, Using constraint programming and local search methods to solve vehicle routing problems, in: Michael Maher, Jean-Francois Puget (Eds.), Principles and Practice of Constraint Programming, CP98, Springer, Berlin, Heidelberg, 1998, pp. 417–431.
[171] Paulo Vitor Silvestrin, Marcus Ritt, An iterated tabu search for the multi-compartment vehicle routing problem, Computers & Operations Research 81 (C) (May 2017) 192–202.
[172] Marius M. Solomon, Algorithms for the vehicle routing and scheduling problems with time window constraints, Operations Research 35 (2) (April 1987) 254–265.
[173] Marius M. Solomon, Jacques Desrosiers, Time window constrained routing and scheduling problems, Transportation Science 22 (1) (1988) 1–13.
[174] Mara Soto, Marc Sevaux, Andr Rossi, Andreas Reinholz, Multiple neighborhood search, tabu search and ejection chains for the multi-depot open vehicle routing problem, Computers & Industrial Engineering 107 (C) (May 2017) 211–222.
[175] Gilbert Syswerda, in: Lawrence Davis (Ed.), Handbook of Genetic Algorithms, Van Nostrand Reinhold, 1991.
[176] W.Y. Szeto, Yongzhong Wu, Sin C. Ho, An artificial bee colony algorithm for the capacitated vehicle routing problem, European Journal of Operational Research 215 (1) (2011) 126–135.
[177] A. Taha, M. Hachimi, A. Moudden, A discrete bat algorithm for the vehicle routing problem with time windows, in: 2017 International Colloquium on Logistics and Supply Chain Management, LOGISTIQUA, April 2017, pp. 65–70.
[178] Éric Taillard, Phillipe Badeau, Michel Gendreau, François Guertin, Jean-Yves Potvin, A tabu search heuristic for the vehicle routing problem with soft time windows, Transportation Science 31 (2) (1997) 170–186.
[179] Éric D. Taillard, Parallel iterative search methods for vehicle routing problems, Networks 23 (8) (1993) 661–673.
[180] Li Jing Tan, Fu Yong Lin, Hong Wang, BFO optimization algorithms for vehicle routing problem with time windows, in: Vehicle, Mechatronics and Information Technologies II, in: Applied Mechanics and Materials, vol. 543, Trans Tech Publications, 6 2014, pp. 1884–1887.
[181] Lijing Tan, Fuyong Lin, Hong Wang, Adaptive comprehensive learning bacterial foraging optimization and its application on vehicle routing problem with time windows, Neurocomputing 151 (2015) 1208–1215.
[182] Christos D. Tarantilis, Solving the vehicle routing problem with adaptive memory programming methodology, Computers & Operations Research 32 (2005) 2309–2327.
[183] Christos D. Tarantilis, Chris T. Kiranoudis, Boneroute: an adaptive memory-based method for effective fleet management, Annals of Operations Research 115 (1–4) (2002) 227–241.
[184] Christos D. Tarantilis, F. Stavropoulou, Panagiotis P. Repoussis, A template-based tabu search algorithm for the consistent vehicle routing problem, Expert Systems with Applications 39 (2012) 4233–4239.
[185] Ehsan Teymourian, Vahid Kayvanfar, G.H.M. Komaki, M. Zandieh, Enhanced intelligent water drops and cuckoo search algorithms for solving the capacitated vehicle routing problem, Information Sciences 334–335 (2016) 354–378.
[186] S.R. Thangiah, K.E. Nygard, P.L. Juell, Gideon: a genetic algorithm system for vehicle routing with time windows, in: [1991] Proceedings. The Seventh IEEE Conference on Artificial Intelligence Application, vol. i, Feb 1991, pp. 322–328.
[187] Sam R. Thangiah, Ibrahim H. Osman, Tong Sun, Hybrid genetic algorithm, simulated annealing and tabu search methods for vehicle routing problems with time windows, 1993.
[188] Sebastian B. Thrun, Efficient Exploration in Reinforcement Learning, Technical report, Pittsburgh, PA, USA, 1992.
[189] Om Verma, Rashmi Jain, Vindhya Chhabra, Solution of travelling salesman problem using bacterial foraging optimisation algorithm, International Journal of Swarm Intelligence 1 (01 2014) 179–192.
[190] Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, Nadia Lahrichi, Walter Rei, A hybrid genetic algorithm for multidepot and periodic vehicle routing problems, Operations Research 60 (3) (2012) 611–624.

[191] Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, Christian Prins, A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows, Computers & Operations Research 40 (1) (2013) 475–489.

[192] Daniele Vigo, A heuristic algorithm for the asymmetric capacitated vehicle routing problem, European Journal of Operational Research 89 (1) (1996) 108–126.

[193] Juan G. Villegas, Christian Prins, Caroline Prodhon, Andrés L. Medaglia, Nubia Velasco, A grasp with evolutionary path relinking for the truck and trailer routing problem, Computers & Operations Research 38 (9) (2011) 1319–1334.

[194] Chao Wang, Fu Zhao, Dong Mu, John W. Sutherland, Simulated annealing for a vehicle routing problem with simultaneous pickup-delivery and time windows, in: Vittal Prabhu, Marco Taisch, Dimitris Kiritsis (Eds.), Advances in Production Management Systems. Sustainable Production and Service Supply Chains, Springer, Berlin, Heidelberg, 2013, pp. 170–177.

[195] Chao Wang, Shengchuan Zhou, Yang Gao, Chao Liu, A self-adaptive BAT algorithm for the truck and trailer routing problem, Engineering Computations 35 (01 2018).

[196] Junling Wang, Arun Kumar Ranganathan Jagannathan, Xingquan Zuo, Chase C. Murray, Two-layer simulated annealing and tabu search heuristics for a vehicle routing problem with cross docks and split deliveries, Computers & Industrial Engineering 112 (2017) 84–98.

[197] Shuihua Wang, Zeyuan Lu, Ling Wei, Genlin Ji, Jiquan Yang, Fitness-scaling adaptive genetic algorithm with local search for solving the multiple depot vehicle routing problem, Simulation 92 (7) (2016) 601–616.

[198] Lijun Wei, Zhenzhen Zhang, Defu Zhang, Stephen C.H. Leung, A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints, European Journal of Operational Research 265 (3) (2018) 843–859.

[199] Lijun Wei, Zhenzhen Zhang, Defu Zhang, Andrew Lim, A variable neighborhood search for the capacitated vehicle routing problem with two-dimensional loading constraints, European Journal of Operational Research 243 (3) (2015) 798–814.

[200] Shuyuan Yang, Min Wang, Licheng Jiao, A quantum particle swarm optimization, in: Proceedings of the 2004 Congress on Evolutionary Computation, IEEE Cat. No. 04TH8753, vol. 1, 2004, pp. 320–324.

[201] Xin-She Yang, Nature-Inspired Metaheuristic Algorithms, Luniver Press, 2008.

[202] Xin-She Yang, A New Metaheuristic Bat-Inspired Algorithm, Springer, Berlin, Heidelberg, 2010, pp. 65–74.

[203] Baozhen Yao, Ping Hu, Mingheng Zhang, Shuang Wang, Artificial bee colony algorithm with scanning strategy for the periodic vehicle routing problem, Simulation 89 (6) (2013) 762–770.

[204] Baozhen Yao, Qianqian Yan, Mengjie Zhang, Yunong Yang, Improved artificial bee colony algorithm for vehicle routing problem with time windows, PLoS ONE 12 (9) (09 2017) 1–18.

[205] Baozhen Yao, Bin Yu, Ping Hu, Junjie Gao, Mingheng Zhang, An improved particle swarm optimization for carton heterogeneous vehicle routing problem with a collection depot, Annals of Operations Research 242 (2) (Jul 2016) 303–320.

[206] Peng-Yeng Yin, Ya-Lan Chuang, Adaptive memory artificial bee colony algorithm for green vehicle routing with cross-docking, Applied Mathematical Modelling 40 (21) (2016) 9302–9315.

[207] Vincent F. Yu, A.A.N. Perwira Redi, Yosi Agustina Hidayat, Oktaviyanto Jimat Wibowo, A simulated annealing heuristic for the hybrid vehicle routing problem, Applied Soft Computing 53 (C) (April 2017) 119–132.

[208] Emmanouil E. Zachariadis, Christos D. Tarantilis, Chris T. Kiranoudis, A guided tabu search for the vehicle routing problem with two-dimensional loading constraints, European Journal of Operational Research 195 (3) (2009) 729–743.

[209] Shuzhu Zhang, C.K.M. Lee, K.L. Choy, William Ho, W.H. Ip, Design and development of a hybrid artificial bee colony algorithm for the environmental vehicle routing problem, Transportation Research Part D: Transport and Environment 31 (2014) 85–99.

[210] Hongqiang Zheng, Yong-Quan Zhou, Qifang Luo, A hybrid cuckoo search algorithm-grasp for vehicle routing problem, Journal of Convergence Information Technology 8 (02 2013) 821–828.

[211] Yongquan Zhou, Qifang Luo, Jian Xie, Hongqing Zheng, A Hybrid BAT Algorithm With Path Relinking for the Capacitated Vehicle Routing Problem, Springer International Publishing, Cham, 2016, pp. 255–276.

[212] Marcin Woch, Piotr Lebkowski, Sequential simulated annealing for the vehicle routing problem with time windows, Decision Making in Manufacturing and Services 3 (2009).

[213] V. Černý, Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm, Journal of Optimization Theory and Applications 45 (1985) 41–51.