Diversity-Driven Selection of Multiple Crossover Operators for the Capacitated Arc Routing Problem

Pietro Consoli and Xin Yao

CERCIA, School of Computer Science
University of Birmingham
Birmingham, West Midlands
B15 2TT, UK
{p.a.consoli,x.yao}@cs.bham.ac.uk

Abstract. The Capacitated Arc Routing Problem (CARP) is a NP-Hard routing problem with strong connections with real world problems. In this work we aim to enhance the performance of MAENS, a state-of-the-art algorithm, through a self-adaptive scheme to choose the most suitable operator and a diversity-driven ranking operator. Experimental results on 181 problem instances show how these techniques can both improve the results of the current state-of-the-art algorithms and provide good directions to develop EAs with a more robust approximation ratio.

Keywords: Memetic Algorithm, Stochastic Ranking, Capacitated Arc Routing Problem, Self-Adaptation, Approximation Algorithms

1 Introduction

The *Capacitated Arc Routing Problem* (CARP) 1981[1] has been an object of study mostly because of its close relationship with real world problems. For the CARP a considerable number of exact methods, heuristics and meta-heuristics has been created. Among the meta-heuristics, several population-based approaches can be mentioned. Lacomme et al. proposed a Genetic Algorithm[2], a Memetic Algorithm[3] as well as an ant-based approach[4]. Chu et al. proposed a scatter search in 2006[5] while Beullens et al. proposed a Guided Local Search [6] using a heuristic based on the evaluation of the potential mutations of each solution. Hertz and Mittaz[7] used a Variable Neighbourhood Search which uses a set of new mutation operators.
Tang et al. proposed a Memetic Algorithm[8] named MAENS whose local search is provided with a long-step operator, named Merge-Split.
Several algorithms have been proposed for different versions of the problem, such as the Memetic Algorithm in [9] for the Periodic CARP. Mei et al. also proposed an approach for the multi objective CARP in [10]. Another attempt based on a Tabu Search is described in [11]. Xing et al. developed in [12] a hybrid algorithm based on the ACO meta-heuristic for the Extended CARP, which introduces several constraints, such as a maximum service time, penalties for turns

Table 1. Summary of CARP definition. $app(S_i)$ counts the times that a task appears in the sequence S, $inv(S_i)$ returns the task in the opposite direction of S_i and $nveh$ is the number of maximum vehicles allowed

Problem Definition	$\min \mathrm{TC}(S) = \sum_{i=1}^{\mathrm{length}(S)-1}(sc(S_i) + sp(S_i, S_{i+1}))$
Constraints	$load(R_i) \leq Q$ $app(S_i) = 1, \forall S_i \in A_R$ $m <= nveh$
Total Service Cost	$\mathrm{TC} = \sum_{i=1}^{\mathrm{length}(S)-1}(sc(S_i) + sp(S_i, S_{i+1}))$
Total Load	$load(R_k) = \sum_{i=1}^{\mathrm{length}(R_k)} d(S_{ik})$

and a variable amount and position of depots. In [13] Xing et al. proposed an evolutionary approach for the Multi-Depot CARP. More recently, Mei et al[14] adopted a Cooperative Co-evolution framework[15] to decompose Large Scale CARP instances.
For NP-Hard problems strongly connected with real world applications as the CARP, it is important that algorithms that we design are reliable enough to return good solutions, when not optimal, in as many cases as possible. However, for this problem, the work on the reinforcement of the average results of the algorithms has been lacking.
The aim of this research is therefore to strengthen the approximation ability of a state-of-the-art algorithm for the CARP through the following techniques:

- the use of a novel distance measure between CARP solutions;
- a diversity-driven stochastic ranking operator;
- a set of new recombination operators for the problem;
- an Adaptive Operator Selection strategy which uses a new Credit Assignment technique based on the aforementioned ranking operator.

The rest of the paper is organized as follows. Section 2 introduces some preliminary notions such as the definition of the CARP and the MAENS algorithm which represents the state-of-the-art for this problem. Section 3 presents a novel diversity measure for CARP solutions and a diversity-driven ranking operator. Section 4 introduces a suite of crossover operators for the problem and the Adaptive Operator Selection technique adopted. Section 5 shows the performance of the MAENS algorithm when using the proposed techniques. Section 6 includes the conclusions and the future work.

2 Background

2.1 Problem Definition

$G = (V, A)$ is a connected directed graph where V is the set of vertices, A is the set of arcs, and the subset $A_R \subset A$ is the subset of *required arcs*. Elements of

A are also called *tasks*, while A_R will be the *required tasks*. Each tasks t has a *demand* $d(t)$ which indicates the load necessary to serve the task, a *service cost* $sc(t)$ of crossing the task, a *dead-heading cost* $dc(t)$ of crossing the task without serving it, an *ID* and a reference to their *head(t)* and their *tail(t)* task. The *tasks* need to be served by a fleet of m vehicles with a capacity C and whose route starts and ends in a vertex called *depot*. Each task must be served within a single tour and each vehicle is bound to its capacity. A solution for a CARP instance can therefore be represented by a set of routes, which are sequences of tasks that need to be visited in the given order. To distinguish between routes a *dummy task* is commonly used with *ID* 0, which represents the vehicle being in the depot with null *service cost* and *demand*. The objective is therefore to minimize the total service cost (TC) of the routing subject to the previously mentioned constraints.
The TC is calculated in the following way. When the vehicle is serving a *required task* S_i, the TC will include its serving cost $sc(S_i)$ plus the total cost of the shortest path (*sp*) necessary to connect the *tail* of the task to the *head* of the next task S_{i+1}, obtained through the use of Dijkstra algorithm[16]. A full definition of the problem is included in Table 1.

2.2 MAENS

The Memetic Algorithm with Extended Neighbourhood Search (MAENS)[8] is one of the most competitive and efficient algorithms in the context of the CARP. Its pseudo-code is provided in figure 1. It is possible to divide it into four phases: initialization (lines 1-7), crossover (lines 10-11), local search (line 14) and stochastic ranking (line 26). During the initialization phase, solutions are generated through the use of the Path-Scanning procedure [1]. In the main cycle, couples of parent solutions are randomly selected to generate an offspring using the Sequence Based Crossover (SBX) operator [17]. A local search is therefore performed on the neighbourhood of the solution, with a probability *lsprob*, using three move operators, namely Single Insertion (one task is moved to another route), Double Insertion (two consecutive tasks are moved to another route) and Swap (two tasks of different routes exchange their places). The best solution is then improved through the Merge-Split operator, which applies the Path Scanning procedure[1] and the Ulusoy Splitting tour [18] to the tasks of two randomly

```
1: initialise a population pop;
2: while (pop not full OR attempts < trial) do
3:     generate a new individual p;
4:     if (p is not a clone) then
5:         add p to pop;
6:     end if
7: end while
8: while (stopping criterion is not satisfied) do
9:     for (i = 0 to size*offset) do
10:        randomly select p_1, and p_2 from pop;
11:        generate s^i_x = SBX(p_1, p_2);
12:        extract a random value r
13:        if (r < lsprob) then
14:            generate s^i_ls = LocSearch(s^i_x,p);
15:        else
16:            s^i_ls = s^i_x;
17:        end if
18:        if (s^i_ls is a clone of a parent) then
19:            overwrite parent;
20:        else
21:            if (s^i_ls not a clone) then
22:                add to pop;
23:            end if
24:        end if
25:    end for
26:    pop = Stochastic Ranking(pop);
27: end while
```

Fig. 1. MAENS Algorithm

selected routes. The local search procedure is completed by another iterative search using the three move operators. The offspring are then compared to the solutions of the previous generation and sorted using the Stochastic Ranking procedure [19].

Analysis of the algorithm We have analysed the algorithmic features of MAENS[8] in order to identify what might be the possible drawbacks that affect the robustness of its results. As a memetic algorithm, it is equipped with a local search operator which greatly improves its capacity of exploiting the good solutions found during the search. This exploitation ability might not be balanced enough by an efficient exploration as the algorithm does not have any mechanism that maintains the diversity of the population. The algorithm makes use of a single heuristic, the Path-Scanning[1], in both the initialization phase and within the local search. This might affect the ability of the algorithm to generate new routes during this phase. We choose therefore to contrast the exploitation ability of the algorithm embedding a control over the diversity of the population in the ranking operator and increasing the breadth of the recombination move replacing the SBX with a suite of different operators.

2.3 Approximation Algorithms

An Approximation Algorithm can be defined as follows[20]. Given a minimization optimization problem P with a cost function c, and an algorithm A that is able to produce a feasible solution $f_A(I)$ for the instance I of P whose optimal solution is $\overline{f}(I)$, then the algorithm A is an $\epsilon-$approximate algorithm of P if for any instance I of P :

$$\epsilon \geq \frac{|c(f_A(I)) - c(\overline{f}(I))|}{c(\overline{f}(I))}$$

for some value $\epsilon \geq 0$ and ϵ is the *approximation ratio* of A. As explained in [21], when evaluating the approximation of EAs, due to their stochastic nature, it would be more convenient to evaluate the estimated value $E(f_A(I))$ instead of the results obtained by a single execution of the EA.

3 A Distance Measure for the CARP

In order to be able to evaluate the performance of the algorithm in terms of exploration ability, it is necessary to define a distance measure between solutions. We have identified three possible approaches to define such a measure and we propose a novel one.

3.1 Measuring the Average Diversity of the Population

A first approach is to consider routes as clusters of tasks and consequently to choose an index that is commonly used in the data clustering context. Two non

trivial issues affect this approach: it does not take into account the task service order and it has a high computational cost ($O(n^2)$).
One could choose a different approach by considering routes as strings and to use similarity indexes taken from this field, such as the Levenshtein distance [22]. Although this approach is more precise than the previous one, it is still computationally non trivial ($O(n^2)$) and it has the issue of depending on the chosen representation.
A third approach is that of considering the relationship of consecutiveness between edges as items of the dataset (e.g. if task t_2 follows task t_1 the couple (t_1, t_2) is an item of the dataset) and using a similarity measure between sets. The task service order is therefore split into couples of tasks. This approach has a linear computational time ($O(n)$) and it achieves a higher precision, if compared to the classic clustering approach, by taking into account the task service order. Such a measure has been successfully used in the context of multi-objective Vehicle Routing Problem as in [23] where the number of overlapping arcs is measured through the use of the Jaccard Index[24]. However, since this measure is not a proper metric, despite being valid to perform comparisons between solutions, it might be not fit to compute average distances. For this reason we propose a new distance measure able to deal with this issue.

3.2 A Revised Distance Measure Based on Neighbour Tasks

```
find p1(t) and n1(t)∀t in S1;
find p2(t) and n2(t)∀t in S2;
for (each task t) do
  if (task t is served in both solutions) then
    if ( p1(t) = p2(t) ) then
      add one;
    end if
    if ( n1(t) = n2(t) ) then
      add one;
    end if
  end if
end for
divide the obtained value by 2N.
```

Fig. 2. Diversity Measure for CARP

Similarly to the aforementioned measure, we exploit the relation of consecutiveness between tasks inside the routes. However, the similarity measure adopted in this work is not based on the number of shared arcs between two tasks, but on the number of tasks which share their previous or next arcs. This approach maintains the linear computational cost and takes into account the service order, but has the advantage of always having consistent measurements as the number of tasks is a constant value. The proposed metric between two solutions S_1 and S_2 is described in figure 3.2.
We define $p_i(t)$ and $n_i(t)$ as the functions that return respectively the previous and the next tasks of task t in solution i. Clearly this similarity measure will be equal to 0 if the two solutions are completely different (when $p_1(t) \neq p_2(t)$ and $n_1(t) \neq n_2(t), \forall t$) and equal to 1 if the solutions are identical ($p_1(t) = p_2(t)$ and $n_1(t) = n_2(t), \forall t$). We also point out as the $2N$ possible values achievable are uniformly distributed in the $[0 - 1]$ space, this allows the calculation of average similarities within the population.

3.3 Diversity-Driven Stochastic Ranking

We have initially used such a measure to define a new ranking operator with the aim to preserve the diversity of the population.
In order to do that, we have embedded this information in the stochastic ranking operator[19], which has also been used in the MAENS algorithm[8]. The pseudo-code of the Diversity-Driven Stochastic Ranking is included in Figure 3.3. The main difference with the original algorithm is in line 4. While in the stochastic ranking the comparison based on the fitness value of the solutions is performed either when they both have no constraint violation or with a probability R (usually between 0.40 and 0.50), in our version the comparison is performed when the average distance of the solution p from all the rest of the population, called d_{AVG} is less than that of solution q. Therefore, we consider the value d_{AVG} as a *crowding distance* which is supposed to bias the search towards the solutions which lie in areas of the search space which have not been thoroughly exploited. As this is the first attempt to use such a measure, the work might be extended by using a more refined one that might consider either only the n closest solutions, or the number of solutions which are at least λ-distant or also the distance from the *centroid* individual of the population.

```
1:  for (size of the population) do
2:    for (each individual p in the population) do
3:      extract a random value r;
4:      if (d_AVG(p) < d_AVG(q) OR r < R) then
5:        if (fitness(p)>fitness(q) then
6:          switch position of p and q;
7:        end if
8:      else
9:        if (fitness(p)>fitness(q) then
10:         switch position of p and q;
11:       end if
12:     end if
13:   end for
14: end for
```

Fig. 3. Diversity-Driven Stochastic Ranking

4 Operator Selection

As previously mentioned, the use of a single heuristic to generate routes might limit the exploration capacity of the algorithm and consequently its ability to escape local optima.
We propose the use of a suite of crossover operators under the assumption that it might lead to a more robust performance of the algorithm.
This offers the advantage to use each heuristic in those instances where they perform better. As this is not known a priori, it is necessary to define a operator-selection strategy which is able to address the problem of selecting the best performing crossover operator for each instance. Besides, the use of the proper combination of heuristics might improve further the performance of the single use of each one of them, as different heuristics might be the best in different phases of the search.

4.1 Crossover Operators

We have defined four new crossover operators for the CARP problem, namely GSBX, GRX, PBX, SPBX, which are described in this section.

Greedy Sequence Based Crossover (GSBX) The first operator applies a greedy selection to the SBX[17] operator. A route from each solution is randomly selected. These routes are therefore split in two parts and combined in order to generate two different solutions.
The greedy choice replaces the random selection of routes using the following rule that supports the selection of those routes which might have been less efficiently filled. Thus, the route A will have a higher probability to be chosen than route B if $Q - \text{load}(A) \leq Q - \text{load}(B)$ where Q is the maximum load that a vehicle can carry.
Greedy Route Crossover (GRX) The GRX operator adapts the concept of the GPX operator for the Graph Colouring Problem [25] in the context of the CARP. With a round robin criterion, the best route in one of the parent solutions is selected. The selection is performed with the same greedy-rule used in the GSBX operator.
The route is then copied in the offspring and its tasks are removed from parent solution routes. This process is repeated until all remaining routes R in parent solutions have $\text{load}(R) < Q/2$. The remaining routes are therefore randomly selected and combined to form new routes or inserted (when $|R| = 1$) in the existing ones in the positions which minimize the total service cost of the solution.
Pivot Based Crossover (PBX) The PBX operator ranks a list of tasks in a similar way to the Augment-Insert[26] heuristic, although it differs from it as the tasks are ranked according to the load of their outward and return paths, instead of their total service cost.
A route from each parent individual is randomly selected and their tasks are inserted in a unordered list L. A *pivot* task is therefore identified by choosing with higher probability if either its outward or return path has the highest load when using only tasks $t \in L$. The selected subpath is adopted and the rest of the route is built by choosing to insert the tasks in L that minimize the total cost of the route and that do not violate the capacity constraint.
Shortest Path Based Crossover (SPBX) The SPBX applies the idea of the PBX operator to a couple of tasks (t_A, t_B). If the total cost of the shortest path using the tasks of a list L is $SP_L(\cdot, \cdot)$, the algorithm will select with a higher probability the couple t_A and t_B belonging to L such that $SP_L(t_A, t_B)$ is maximised while $SP_L(depot, t_A)$ and $SP_L(t_B, depot)$ are minimised.

4.2 Adaptive Operator Selection

The Adaptive Operator Selection (AOS) is typically composed of an *Operator Selection Rule* (OSR) , such as [27, 28], and a *Credit Assignment Mechanism.*
The Operator Selection scenario can be seen as a dynamic version of the game theory Multi Armed Bandit problem. Each operator represents an arm with an unknown reward probability. The aim is therefore to select during each timestep the arm that maximises the probability of obtaining the reward. Several experiments have shown how MAB-based approaches outperform the former techniques[29].

Common Credit Assignment techniques rely on the evaluation of the fitness function, assigning a higher reward to the operator whose offspring shows the higher improvement. Some reward techniques which also consider diversity have been proposed in the context of multi-modal problems, such as [30]. We have chosen therefore to adopt a MAB-based technique, called dMAB, first proposed in [29], and we have combined it with a new credit assignment technique based on the use of the diversity-driven ranking operator that we have previously introduced. The choice of dMAB is merely dictated by the fact that it is one of the most promising techniques in the area of AOS, as in this stage we are not interested into identifying the most efficient operator. Comparisons of the performance of dMAB with respect other techniques, which have not been performed in this context, can be found in dMAB original paper[29].

Dynamic Multi-Armed Bandit The dMAB adapts the classic Multi-Armed Bandit scenario to a dynamic context where the reward probability of each arm is not independent and is not fixed. To address the dynamic context problem, the classical Upper Confidence Bound (UCB1)[31] algorithm is combined with a Page Hinkley test[32], to identify the change of reward probabilities. More information can be found in the original paper [29].

Credit Assignment by Stochastic Ranking We propose a new credit assignment mechanism exploiting the selection operated by the diversity-driven ranking operator. We assign a reward r which is proportional to the number of offspring generated by the selected operator that will survive to the next generation after the stochastic ranking operator has been applied. Therefore, $r = 0$ when none of the individuals generated by the selected operator has survived the ranking process and $r = 1$ when only individuals generated by it have been chosen by the ranking operator.
Such a technique shows several advantages with respect to classic fitness-based credit assignment ones:

- it takes into account the fitness of the solutions, their similarity and the violation of the constraints;
- the adaptive operation selection process does not require domain knowledge;
- the reward values are always normalized and there is no need to derive a scaling factor;
- unlike fitness-based techniques, it is not affected by the convergence speed of the algorithm.

5 Experimental Studies

We have tested the results of the original algorithm against several versions that we have labelled *MAENSd*, *MAENSm* and *MAENS**, which are respectively the versions of the algorithm adopting the diversity-driven stochastic ranking

Name	Description	Value
psize	population size	30
ubtrial	maximum attempts to generate each initial solution	50
opsize	offspring generated during each generation	6*psize
P_{ls}	probability of performing the local search	0.2
p	routes selected during MergeSplit	2
G_m	maximum generations	500
SR_r	probability of sorting solutions according to their fitness	0.45
σ	tolerance factor for Page-Hinkley test	0.05
λ	change threshold for Page-Hinkley test	1.25

Table 2. MAENS* parameters

operator, the one using the proposed AOS strategy and the combination of both techniques.

The comparison has been carried out on four benchmark test sets, namely *gdb*[33] (23 instances), *val*[34] (34 instances), *egl*[35] (24 instances), and *Beullens et al.*[6], which is composed of four groups of 25 instances, namely *C*,*D*,*E* and *F*. The Wilcoxon Signed-Rank test[36] has been used to perform a statistical hypotesis test between MAENS and each of the proposed versions. The test has been conducted using the *R* software environment[37]. In each case, the test has rejected the null hypothesis that the results of the two algorithms were not significantly different. Table 4 reports the details of such tests.

	MAENS	MAENSd	MAENSm	MAENS*
W	–	85	79	92
D	–	77	74	76
L	–	19	28	13
avg	2040.77	2035.28	2035.90	2033.20
std	9.42	4.91	6.37	4.52
best	2026.25	2027.29	2026.09	2025.82
ϵ	.0195	.0158	.0164	.0151

Table 3. A summary of the results of the four algorithms. Each column shows the number of instances where each algorithm achieved a better average fitness (W), performed equally (D) or worse (L) than MAENS, the mean average fitness (avg), standard deviation (std) and best result (best) as well as the mean approximation ratio (ϵ)

Table 2 shows the parameters used to execute the algorithm for 30 independent trials on each of the 181 instances. The algorithm has not been through a process of parameter configuration. Parameters present in the original version of the algorithm (first 7 parameters included in table 2) have kept their original values. New parameters such as those necessary for the Page-Hinckley test have been identified through a few test-and-trial attempts, using the set of benchmark instances as a training set.

For the sake of brevity we do not include the complete results of the comparisons, which are however available on the authors website[1], but we only report a summary of the results in table 3. In terms of mean average fitness, both MAENSd and MAENSm manage to outperform the original algorithm in 85 and 79 instances, and lose the comparison only in 19 and 28 cases. The results of their

	MAENSd	MAENSm	MAENS*
V	4683	4917	5238.5
p-values	2.426e-10	2.945e-10	4.164e-15

Table 4. Results of two-sided tests of significance for the three proposed versions of the MAENS algorithm. The columns show the V statistic computed with the Wilcoxon Signed-Rank Test and the p-value obtained

[1] http://www.cs.bham.ac.uk/~pac265/

combined version MAENS* confirm how the combination of the two techniques has a constructive effect on the algorithm, achieving a better average fitness in 92 instances and losing in only 13. In terms of average standard deviation, the results of MAENSd and MAENSm represent an improvement with respect to the original algorithm, lowering it from 9.42 to 4.91 in the first case and to 6.37 in the second case. Even in this case the results of MAENS* improve the result achieving an average standard deviation of 4.52. We interpret this result as a sign of an improved convergence reliability. The average mean results obtained by the three algorithms reflects the previously mentioned analysis.
An interesting result is that of the average best result obtained by the algorithms. MAENSd showed a somehow predictable reduction of the algorithm exploitation ability (2027.29 against 2026.25 of MAENS) while MAENSm managed to slightly improve the average best result (2026.09). As a consequence, it would be reasonable to expect MAENS* results to lie between those of MAENSd and MAENSm. However, MAENS* achieved the best results (2025.82), confirming the effectiveness of this combination, which discovered new optima for 15 instances.
We have also compared the algorithms in terms of their average approximation-ratio. MAENS ratio of 0.0195 has been reduced to 0.0158 and 0.0164 in the cases of MAENSd and MAENSm and to 0.0151 in the case of MAENS*.
With regard to the runtime, MAENS* is essentially comparable to its original version. The additional computational cost introduced by the calculation of the average diversity of the solutions is balanced by the improved convergence speed, while the AOS does not add any noticeable cost in the algorithm, whose computational cost is still largely dominated by the local search procedure.

6 Conclusions

We have proposed an improved version of the current state-of-the-art algorithm for the CARP[8], called MAENS*. The main characteristics of this algorithm are (a) a new diversity measure between solutions, (b) a diversity-driven stochastic ranking operator, (c) an AOS strategy using four novel crossover operators and (d) a novel Credit Assignment strategy using the aforementioned ranking operator to define rewards. The results of a comparison between the two algorithms show how MAENS* outperformed the the original algorithm in terms of average fitness and produced more robust and reliable results.
The work carried out so far leaves space to several possible improvements, as optimal values of the parameters adopted could be identified, as well as generalizations of the techniques proposed in this work to other combinatorial optimisation problems.

Acknowledgement
This work was supported by EPSRC (Grant No. EP/I010297/ 1). Xin Yao was supported by a Royal Society Wolfson Research Merit Award. The authors would like to thank the anonymous reviewers for their insightful and constructive comments.

References

1. Golden, B.L., Wong, R.T.: Capacitated arc routing problems. Networks **11**(3) (1981) 305–315
2. Lacomme, P., Prins, C., Ramdane-Chérif, W.: A genetic algorithm for the capacitated arc routing problem and its extensions. In: Applications of evolutionary computing. Springer (2001) 473–483
3. Lacomme, P., Prins, C., Ramdane-Cherif, W.: Competitive memetic algorithms for arc routing problems. Annals of Operations Research **131**(1-4) (2004) 159–185
4. Lacomme, P., Prins, C., Tanguy, A.: First competitive ant colony scheme for the carp. In: Ant Colony Optimization and Swarm Intelligence. Springer (2004) 426–427
5. Chu, F., Labadi, N., Prins, C.: A scatter search for the periodic capacitated arc routing problem. European Journal of Operational Research **169**(2) (2006) 586–605
6. Beullens, P., Muyldermans, L., Cattrysse, D., Van Oudheusden, D.: A guided local search heuristic for the capacitated arc routing problem. European Journal of Operational Research **147**(3) (2003) 629–643
7. Hertz, A., Mittaz, M.: A variable neighborhood descent algorithm for the undirected capacitated arc routing problem. Transportation Science **35**(4) (2001) 425–434
8. Tang, K., Mei, Y., Yao, X.: Memetic algorithm with extended neighborhood search for capacitated arc routing problems. Evolutionary Computation, IEEE Transactions on **13**(5) (2009) 1151–1166
9. Mei, Y., Tang, K., Yao, X.: A memetic algorithm for periodic capacitated arc routing problem. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on **41**(6) (2011) 1654–1667
10. Mei, Y., Tang, K., Yao, X.: Decomposition-based memetic algorithm for multiobjective capacitated arc routing problem. Evolutionary Computation, IEEE Transactions on **15**(2) (2011) 151–165
11. Mei, Y., Tang, K., Yao, X.: A global repair operator for capacitated arc routing problem. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on **39**(3) (2009) 723–734
12. Xing, L.N., Rohlfshagen, P., Chen, Y.W., Yao, X.: A hybrid ant colony optimization algorithm for the extended capacitated arc routing problem. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on **41**(4) (2011) 1110–1123
13. Xing, L., Rohlfshagen, P., Chen, Y., Yao, X.: An evolutionary approach to the multidepot capacitated arc routing problem. Evolutionary Computation, IEEE Transactions on **14**(3) (2010) 356–374
14. Mei, Y., Li, X., Yao, X.: Cooperative co-evolution with route distance grouping for large-scale capacitated arc routing problems. IEEE Transactions on Evolutionary Computation, accepted on 31 July 2013 (2013)
15. Potter, M.A., De Jong, K.A.: A cooperative coevolutionary approach to function optimization. In: Parallel Problem Solving from NaturePPSN III. Springer (1994) 249–257
16. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische mathematik **1**(1) (1959) 269–271
17. Potvin, J.Y., Bengio, S.: The vehicle routing problem with time windows part ii: genetic search. INFORMS journal on Computing **8**(2) (1996) 165–172

18. Ulusoy, G.: The fleet size and mix problem for capacitated arc routing. European Journal of Operational Research **22**(3) (1985) 329–337
19. Runarsson, T.P., Yao, X.: Stochastic ranking for constrained evolutionary optimization. Evolutionary Computation, IEEE Transactions on **4**(3) (2000) 284–294
20. Papadimitriou, C.H., Steiglitz, K.: Combinatorial optimization: algorithms and complexity. Courier Dover Publications (1998)
21. He, J., Yao, X.: An analysis of evolutionary algorithms for finding approximation solutions to hard optimisation problems. In: Evolutionary Computation, 2003. CEC'03. The 2003 Congress on. Volume 3., IEEE (2003) 2004–2010
22. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions and reversals. In: Soviet physics doklady. Volume 10. (1966) 707
23. Garcia-Najera, A.: Preserving population diversity for the multi-objective vehicle routing problem with time windows. In: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers, ACM (2009) 2689–2692
24. Jaccard, P.: Etude comparative de la distribution florale dans une portion des Alpes et du Jura. Impr. Corbaz (1901)
25. Galinier, P., Hao, J.K.: Hybrid evolutionary algorithms for graph coloring. Journal of combinatorial optimization **3**(4) (1999) 379–397
26. Pearn, W.L.: Augment-insert algorithms for the capacitated arc routing problem. Computers & Operations Research **18**(2) (1991) 189–198
27. Goldberg, D.E.: Probability matching, the magnitude of reinforcement, and classifier system bidding. Machine Learning **5**(4) (1990) 407–425
28. Thierens, D.: An adaptive pursuit strategy for allocating operator probabilities. In: Proceedings of the 2005 conference on Genetic and evolutionary computation, ACM (2005) 1539–1546
29. Da Costa, L., Fialho, A., Schoenauer, M., Sebag, M., et al.: Adaptive operator selection with dynamic multi-armed bandits. In: Genetic and Evolutionary Computation Conference (GECCO). (2008) 913–920
30. Maturana, J., Saubion, F.: A compass to guide genetic algorithms. In: Parallel Problem Solving from Nature–PPSN X. Springer (2008) 256–265
31. Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multiarmed bandit problem. Machine learning **47**(2-3) (2002) 235–256
32. Hinkley, D.V.: Inference about the change-point from cumulative sum tests. Biometrika **58**(3) (1971) 509–523
33. DeArmon, J.S.: A comparison of heuristics for the capacitated Chinese postman problem. PhD thesis, University of Maryland (1981)
34. Benavent, E., Campos, V., Corberán, A., Mota, E.: The capacitated arc routing problem: lower bounds. Networks **22**(7) (1992) 669–690
35. Eglese, R.W.: Routing winter gritting vehicles. Discrete applied mathematics **48**(3) (1994) 231–244
36. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics bulletin **1**(6) (1945) 80–83
37. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. (2013)