CrossMark

Dynamic selection of evolutionary operators based on online learning and fitness landscape analysis

Pietro A. Consoli[1] · Yi Mei[2] · Leandro L. Minku[3] · Xin Yao[1]

Abstract Self-adaptive mechanisms for the identification of the most suitable variation operator in evolutionary algorithms rely almost exclusively on the measurement of the fitness of the offspring, which may not be sufficient to assess the optimality of an operator (e.g., in a landscape with an high degree of neutrality). This paper proposes a novel adaptive operator selection mechanism which uses a set of four fitness landscape analysis techniques and an online learning algorithm, dynamic weighted majority, to provide more detailed information about the search space to better determine the most suitable crossover operator. Experimental analysis on the capacitated arc routing problem has demonstrated that different crossover operators behave differently during the search process, and selecting the proper one adaptively can lead to more promising results.

Keywords Adaptive operator selection · Online learning · Capacitated arc routing problem

✉ Pietro A. Consoli
  p.a.consoli@cs.bham.ac.uk

  Yi Mei
  yi.mei@ecs.vuw.ac.nz

  Leandro L. Minku
  leandro.minku@leicester.ac.uk

  Xin Yao
  x.yao@cs.bham.ac.uk

[1] CERCIA, School of Computer Science, University of Birmingham, Birmingham, West Midlands B15 2TT, UK

[2] Victoria University of Wellington, Wellington 6140, New Zealand

[3] Department of Computer Science, University of Leicester, University Road, Leicester LE1 7RH, UK

1 Introduction

Parameter setting is an important area of research in the evolutionary computation field. Since an a-priori identification of the optimal configuration of the parameters is always time-consuming and often impractical, one must employ a dynamic selection strategy of the optimal configuration which is performed while the search is being executed. In addition, a static set of parameters is not always the optimal choice for a large number of problems where self-adapting techniques have proven to be more effective (Eiben et al. 1999).

The problem of identifying the most suitable variation operator among several, also known as adaptive operator selection (AOS), can be divided into two sub-tasks: the credit assignment (CA) mechanism, used to evaluate the performance of the operators; and the operator selection (OS) rule, necessary to determine the most suitable operator using the information provided by the CA mechanism. The majority of the credit assignment approaches in literature are based on the evaluation of the fitness of the offspring generated by the operator, which is compared either to the current best solution (Davis 1989), to the median fitness (Julstrom 1995) or to the parents' fitness (Barbosa and Sá 2000). A different strategy evaluating both fitness and diversity of the offspring was proposed in Maturana and Saubion (2008). The reward has been mostly considered as the value assessed during the last evaluation (*instantaneous* reward), as the average reward over a window of the last N evaluations (*average* reward), and as the biggest improvement achieved over a window of the last N evaluations (*extreme* reward) (Fialho et al. 2009).

The use of alternative metrics has been recently considered in Soria Alcaraz et al. (2014), where an evolvability metric replaces the evaluation of the fitness. A different approach for population-based meta-heuristics, proposed in Consoli

and Yao (2014), assesses the reward as the proportion of solutions generated by each operator which have been selected by the ranking phase of the evolutionary algorithm. The credit assignment mechanism is coupled with operator selection rules such as probability matching (Goldberg 1990), adaptive pursuit (Thierens 2005) or multi-armed bandit solvers (MAB) (DaCosta et al. 2008). Several improvements of the MAB strategy have been proposed, as in Belluz et al. (2015), Chen et al. (2013) and Kim et al. (2012). Reinforcement learning has been also used in parameter setting (Karafotias et al. 2015), as in Eiben et al. (2007), where a reinforcement learning procedure is adopted to modify the parameters on-the-fly, or in Sakurai et al. (2010) where the selection probability of the operators is adaptively changed using a reinforcement learning approach.

From the analysis of the existing literature, it is clear that almost all the existing CA strategies rely exclusively on the mere evaluation of the fitness of the offspring. However, the information provided by the fitness at a single generation may not be sufficient to assess the optimality of an operator (e.g., in a landscape with a high degree of neutrality). The purpose of our work is therefore to develop a new dynamic CA mechanism which considers a suite of measures, and that can be adopted also as an operator selection rule. We consider the memetic algorithm with extended neighborhood search (MAENS*) (Consoli and Yao 2014) algorithm as a case study and for comparison purposes. More specifically, we aim to answer the following research questions in our paper:

- *RQ1* What kind of additional information we can provide to the credit assignment technique for a more "aware" calculation of the reward and does this information effectively help to improve the prediction ability of the algorithm?
- *RQ2* What technique would be useful to handle such data and to select the most suitable operator in such a dynamic environment? Would the prediction ability of the technique be better than that of MAENS*? Would the use of this technique improve the optimization ability of MAENS*?

The contributions of our work include:

- An ensemble of four different online fitness landscape analysis techniques, performed during the execution of the MAENS* algorithm in order to give a more accurate description of the current population (RQ1).
- A credit assignment technique based on the use of a online learning algorithm to predict the reward of the most suitable operator (RQ2).
- Two different reward measures are studied: one based on the survival ability of the offspring and another one based on the analysis of their diversity.

This work extends our previous work in Consoli et al. (2014) with new experiments and contributions. In particular: (a) we investigate the use of a novel reward measure called diversity-based reward (DBR); (b) we study the adoption of a different operator selection rule, named concurrent strategy (CS); (c) we extend our analysis by testing our algorithms on a dataset of large CARP instances. The results of the experiments carried out show that the proposed approach is able to produce results with comparable solution quality to a state-of-the-art strategy and reveal how in some cases the presence of a set of measures have a beneficial effect on the optimization ability of the AOS.

The rest of the paper is organized as follows. Section 2 introduces the case scenario and the base MAENS* algorithm. Section 3 describes the novel reward measures and operator selection rules investigated in this work. Section 4 describes the ensemble of fitness landscape techniques used in conjunction with the CA mechanism of the MAENS* algorithm. Section 5 describes the online learning algorithm that has been used and adapted for the CA system. Section 6 presents the proposed MAENS*-II algorithm. Section 7 describes the experiments that have been carried out to verify the assumptions of this research and their results. Finally, the last section includes the conclusions and some future work ideas.

2 Background

To investigate over our research questions, we consider the MAENS* algorithm (Consoli and Yao 2014) for the capacitated arc routing problem (CARP) (Golden and Wong 1981), as the case study of this research, as it already utilizes an adaptive operator selection scenario and provides a term of comparison with alternative techniques. The strong relationship between CARP and specific real-world problems, such as winter gritting, waste collection or postal service make this a problem of great interest for the scientific community, and a large number of heuristics, exact methods and meta-heuristics have been proposed for this problem and its many variants. Although the hyper-heuristic proposed in this work is applied to the capacitated arc routing problem, it would be possible to adapt it to different NP-Hard problems by replacing the low level heuristics and by identifying the best fitness landscape analysis metrics that better describe the specific landscapes of the different NP-Hard problem.

2.1 MAENS*

MAENS*, the case study the for this research, extends the memetic algorithm named MAENS (Tang et al. 2009) introduced in 2009. MAENS is a memetic algorithm which makes use of a crossover operator, a local search combining three

local move operators and a novel long move operator called MergeSplit, and a ranking selection procedure called stochastic ranking (SR) (Runarsson and Yao 2000). The major differences between MAENS and MAENS* are: (a) MAENS uses a single crossover operator, whereas MAENS* uses a set of crossover operators, (b) a dynamic MAB mechanism (dMAB) (Fialho et al. 2009) is adopted as an AOS rule, (c) a novel CA mechanism assigns a reward to the operators which is proportional to the number of solutions generated by each operator that "survived" the ranking phase, named proportional reward, (d) the stochastic ranking is improved considering also the diversity of the solutions (dSR) using a (e) novel diversity measure for the CARP search space.

The dMAB (Fialho et al. 2009) approach, adopted in this work, combines the UCB1 algorithm (Auer et al. 2002) with the Page-Hinckley (PH) statistical test (Hinkley 1971) to detect changes in the environment. When the PH test is triggered, the MAB system is restarted and the information gathered in the previous generations is discarded. The MAENS* algorithm represents one case study of our research, as the presence of a suite of crossover operators allows the study of other AOS approaches.

2.2 Capacitated arc routing problem

The capacitated arc routing problem (CARP) can be formally defined as the problem of minimizing the total service cost of a routing plan, given a set **T** of tasks (which corresponds to a subset of the arcs of a graph) and a fleet of **m** vehicles with capacity **C**. Each task **t** has a *service cost* **sc**, a *demand* **d** (the load of the vehicle necessary to service the task), a unique **id**, a reference to its head and tail vertices, and must be served once and entirely within the same route **R_j**. A CARP solution S can be represented as

$$S = \{\{t_0, t_k, ..., t_l, t_0\}, ..., \{t_0, t_p, ..., t_q, t_0\}\}$$

which is a permutation of the whole set of tasks, divided into several routes **R_j**. Each route must start and end in a specific vertex called *depot*. We use a *dummy task* t_0 with null demand and service cost to show the start and the end of a route in the depot. The service cost of a single route is calculated by adding the service cost of all the tasks in the route plus the cost of the shortest path **sp** between each task. The problem can be formally defined as follows:

$$\min TC(S) = \sum_{i=1}^{length(S)-1} (sc(t_i) + sp(t_i, t_{i+1})),$$

subject to the constraints

$$load(R_j) \leq C, \quad app(t_i) = 1 \quad \text{and} \quad \forall t_i \in T, m \leq n_{veh},$$

$$load(R_j) = \sum_{i=1}^{length(R_j)} d(t_{ij}),$$

where $app(t_i)$ gives the number of appearances of tasks t_i in the sequence of the tasks in S and n_{veh} is the number of available vehicles.

2.3 Recombination operators for CARP

As explained in Sect. 2.1, MAENS* uses a set of crossover operator, instead of a single crossover operator. This section describes the four crossover operators introduced in MAENS* to deal with the CARP problem.

2.3.1 Greedy sequence-based crossover (GSBX)

The GSBX operator, can be considered as a variant of the sequence based crossover (SBX) operator. In this case, one route is extracted from each parent solution following a greedy strategy that influences the selection towards those routes whose vehicle is still not full (the total demand of the route is the smallest). Once selected, the two routes are recombined using a one point crossover mechanism. The route generated in this way replaces the original routes of the parents to generate the new offspring. Since problems caused by double servicing of tasks or tasks not being serviced might arise, the solution goes through a repair phase to guarantee its feasibility.

2.3.2 Greedy route crossover (GRX)

In GRX, an offspring is created by alternatively copying the routes of the parents into it. Routes are extracted from the parent solutions, giving higher priority to the routes with a higher quality measure. Tasks that have been inserted into the offspring are consequently removed from both parents, to avoid the double servicing of tasks. The procedure is repeated until the remaining routes in the parents have less than a certain amount of tasks. In that case, the remaining tasks are inserted in the existing routes or merged into new ones.

2.3.3 Pivot-based crossover (PBX)

Two routes are randomly selected from the parent solutions. The PBX operator works by identifying, among the tasks belonging to such routes, the one that is most suitable to be placed in the middle of a route, which is named *pivot*, as it splits the route in two parts. The route is then rebuilt inserting the remaining tasks in the position that minimizes the total service cost of the route. Finally, the offspring is obtained by replacing the original route in one of the two parents. As in

the case of GSBX, the solution goes through a repair phase to guarantee the feasibility of the solution.

2.3.4 Shortest path-based crossover (SPBX)

The SPBX operator works analogously as the PBX operator, except that in this case, the *pivot* is represented by a path between two of the available tasks. The couple of pivoting tasks is selected as the one that the serves the largest number of available tasks along their path and that minimizes the distance between the extremities of the path and the deposit.

3 Adaptive operator selection

As previously mentioned, AOS is conventionally composed of two different sub-tasks: the credit assignment and the operator selection. For the former part, we propose the use of two different reward measures, named proportional reward (PR) and diversity-based reward (DBR). For the latter, we study the performance of two different strategies: a simple instantaneous reward (IR) approach and a concurrent strategy (CS)-based approach.

3.1 Credit assignment

The choice of the proper credit assignment strategy can be fundamental for the performance of the algorithm. As one objective of this work is to evaluate more than just the fitness of the individuals, we adopt two different strategies that involve the evaluation of different measurements. The first one, named proportional reward, was first used in Consoli and Yao (2014) together with a multi-armed bandit approach. For the second case, we develop a novel measure based on the evaluation of the diversity of the offspring, named diversity-based reward.

3.1.1 Proportional reward (PR)

PR (Consoli and Yao 2014) is a measure of the survival ability of the offspring generated by each crossover operator. We assign a reward r, where $r \in [0, 1]$ corresponds to the percentage of the solutions that have survived the selection phase of the algorithm, and are going to become the parent population for its generation. The use of this technique is a way to entrust the algorithm itself for the evaluation of the offspring. In the case of MAENS*, the offspring able to survive the ranking phase are evaluated according to their fitness value, the amount of violation of the constraints and the average pairwise diversity from the other individuals of the population. The performance of the crossover operator is in this case evaluated at the end of the generation: rather than evaluating the individuals as soon as they are generated,

the PR evaluates their performance in a longer period of time (e.g., an iteration). The PR can be formally represented with the following formula:

$$PR(i)^t = \frac{|x_i : x_i \in \text{parent}^{t+1}|}{|\text{parent}^{t+1}|}$$

where i refers to the ith operator, x_i is an individual obtained through the use of operator i, t is the tth generation and parent^{t+1} is the parent population at the $t + 1$ generation. If more than one operator is used during the same generation, the PR can be calculated in the following way:

$$PR(i)^t = \frac{|x_i : x_i \in \text{parent}^{t+1}|}{|\text{offspring}_i^t|}$$

where offspring$_i^t$ is the set of individuals generated using the operator i during the tth generation.

3.1.2 Diversity-based reward (DBR)

In the case of the DBR, we propose an approach that is opposite to that of the PR, as we evaluate the crossover offspring as soon as they have been generated. As one purpose of the crossover operator is that of introducing diversity in the population through the exploration of new areas of the landscape, we adopt a measure of the diversity introduced by the offspring. In particular, for each operator, we want to measure how distant the offspring are from the parent population, and how wide is the area explored. Therefore, we define a parent distance measure

$$P_d(x) = \frac{d(x, p_1) + d(x, p_2)}{2}$$

as the average distance from the offspring x to its parents p_1 and p_2 and we can consequently compute the average parent distance for operator i, $P_d(i)$, by averaging the $P_d(x)$ of all the offspring generated by such operator. To measure the distance between individuals, we adopt the distance measure for CARP developed in Consoli and Yao (2014). The pseudocode of the distance measure is shown in Fig. 1. Since a CARP solution is represented by a sequence of tasks t, split into different routes, we can define $p_i(t)$ and $n_i(t)$ as two functions that return, respectively, the previous and the next tasks of task t in the sequence of solution S_i. A task t has a perfect correspondence in both solutions if its previous and next tasks match. In the most extreme cases, for two solutions S_1 and S_2, the value of the distance measure will be equal to 1 if $p_1(t) = p_2(t)$ and $n_1(t) = n_2(t), \forall t$, and will be equal to 0 if $(p_1(t) \neq p_2(t)$ and $n_1(t) \neq n_2(t), \forall t)$. In the former case, the two solutions are identical, as there is a full correspondence between the $p_i(t)$ and the $n_i(t)$ of both solutions for each task t, while in the latter case the two solutions are

```
 1  find p₁(t) and n₁(t)∀t in S₁;
 2  find p₂(t) and n₂(t)∀t in S₂;
 3  for (each task t) do
 4  |   if (task t is served in both
    |      solutions) then
 5  |   |   if ( p₁(t) = p₂(t) ) then
 6  |   |   |   add one;
 7  |   |   end
 8  |   |   if ( n₁(t) = n₂(t) ) then
 9  |   |   |   add one;
10  |   |   end
11  |   end
12  end
13  divide the obtained value by 2N.
```

Fig. 1 Diversity measure for CARP solutions

completely different. It is important to point out that while the order in which the tasks in each route are serviced is considered, the order of the routes is not. Therefore two solutions are still identical if they have perfectly corresponding routes even if permutated in a different order. In all the other cases, when the correspondence is partial, the diversity measure will consequently assume values within the range [0, 1].

We also define the *coverage* measure of the operator i

$$C_m(i) = \frac{\sum_i \sum_j d(x_a, x_b)}{N_i^2}$$

as the pairwise average distance between any pair of individuals x_a and x_b that have been generated by it, where N_i is the number of individuals generated through the use of operator i. We can compute the DBR of the ith operator in the following way:

$$\text{DBR}(i) = P_d(i) * C_m(i).$$

Similarly to compass (Maturana and Saubion 2008), this credit assignment technique considers the diversity of the offspring as a criterion to evaluate the performance of the operators. However, there a several differences between such approaches. First, the compass approach addresses the evaluation of both the fitness and the diversity while DBR only considers the diversity, being focused on the evaluation of crossover operators exclusively. Secondly, compass makes use of the Hamming distance entropy as in Lardeux et al. (2006) to measure the population diversity, while DBR deals with both the average pairwise distance of the offspring as well as the distance from the parent population using the CARP based diversity measure shown in Fig. 1.

3.2 Operator selection rule (OSR)

The second step of the AOS process is the operator selection rule. The OSR, given the information gained through the use of the credit assignment mechanism, needs to decide what is the most suitable operator and how to use it. A first problem in this context is that of balancing the exploration of all the operators against the exploitation of the most useful one. In other words, while using the operator that has performed the best so far, one wants to verify whether there is another operator that can do better. A second aspect is that of identifying changes during the execution. As the search goes on, the operator that has performed the best so far might not necessarily be the best one afterwards. It is therefore necessary to balance how much of the "history" relative to each operator one most consider to perform the selection.

In this work, we consider two different approaches for the OSR, namely a single operator-based approach named instantaneous reward and a reinforcement learning-inspired one called concurrent strategy.

3.2.1 Instantaneous reward (IR)

In the IR approach, the offspring is produced through the use of only one crossover operator per generation. As offspring and parent populations are merged in an unique population, it is still possible to evaluate all the crossover operators who have generated a solution that is still present in the population. The operator to use in the next generation $(t + 1)$ is consequently selected as the operator op_i that has obtained the largest reward in the current generation (t):

$$op_i^{t+1} = \max_i(\text{RW}(op_i)^t), \ op_i \in \text{operators}$$

given RW() as a reward measure. Those operators having produced more "extreme" improvements (e.g., discovered new optima) with respect to the others, will have a more favourable evaluation that will last for more generations, even when they have not been selected for the current generation.

The information relative to the previous performances of the operator, except for the last iteration, is discarded. IR is, therefore, designed to be more sensitive to changes, having a bias on the performance of the operators during the previous generation. Finally, the adoption of such approach has the potential risk of eliminating completely an operator from the competition if none of its offspring are present in the current population.

3.2.2 Concurrent strategy (CS)

One of the disadvantages of adopting the instantaneous reward strategy is that it is not possible to identify changes in the environment when only one operator is used. A different approach, therefore, is that of allowing the use of all the operators during all the generations. Such approach, named CS, aims to maximize the gain obtained by using the best

performing operator, and thus allowing the generation of a larger fraction of the offspring by it, while the remaining part is still generated by the other operators. The CS is similar in its behaviour to the adaptive pursuit (AP) approach (Thierens 2005) in its intent to maintain a minimum percentage of the solutions to be generated by the less performing operators. The formula to assign the Selection Rate to each operator i, is the following:

$$\mathrm{SR}_i = \mathrm{SR}_{\min} + (1 - n \times \mathrm{SR}_{\min}) \frac{e^{\mathrm{RW}(op_i)^t/\psi}}{\sum_{j=1}^n e^{\mathrm{RW}(op_j)^t/\psi}}$$

where $\mathrm{SR}_{\min}$ is the minimum selection rate, n is the number of operators, $\mathrm{RW}(op_i)$ is the reward calculated for the operator op_i during the generation t, and ψ is a control parameter that regulates how quickly the system reacts to the changes in the environment. In this case, $n = 4$ since four operators are available.

4 Online fitness landscape analysis

The existing fitness landscape analysis (FLA) techniques have been analysed with the purpose to identify those that can be used in the CARP context. Such selection has been driven by both the necessity to reduce the computational effort by exploiting some calculations that are already performed by the algorithm, and the necessity to identify measures able to "capture" different features of the landscape. We identified four FLA techniques, consisting of one evolvability measure, two neutrality measures and one fitness distribution measure, to describe different features of the landscape and without much increasing the computational effort. The computation of such techniques is based on the evaluation of the neighbourhood of each solution. Such neighbourhood is already generated through the initial iteration of the local search operator of the MAENS algorithm, using the three different move operators involved in this process (single insertion, double insertion, swap insertion). The FLA techniques are employed during each generation, and their results are used as input features of an online learning algorithm to predict the value of one of the two reward measures introduced in Sect. 3.1, to create a more accurate and informative "snapshot" of the current population which eventually might lead to a better selection of the crossover operator. A final remark is necessary about the constraints handling and how it affects the fitness of the individuals. The landscape in which MAENS* operates is that of solutions which may potentially violate the capacity constraints of the vehicles. Therefore, we consider the following fitness function, adopted from Tang et al. (2009):

$$f(S) = \mathrm{TC}(S) + \lambda * \mathrm{TV}(S)$$

where λ is an adaptive parameter depending on the cost, on the violation and on the best feasible solution found so far, $\mathrm{TC}(S)$ is the total cost of the solution and $\mathrm{TV}(S)$ its total violation.

The rest of this section will introduce the four FLA techniques that have been considered in this work and how they are integrated in the MAENS* algorithm.

4.1 Accumulated escape probability

The accumulated escape probability (Lu et al. 2011) is a metric that aims to measure the evolvability, which can be defined as the capacity of the solutions to evolve into better solutions. The accumulated escape probability is obtained by averaging the mean escape rate (Merz 2004) (the proportion of solutions with equal or better fitness in the neighbourhood) of each fitness level with the formula:

$$\mathrm{aep} = \frac{\sum_{f_i \in F} P_j}{|F|}, \quad \text{where } F = f_0, f_1, ..., f_L$$

where f_i is a fitness level (subset of all the solutions with fitness equal to a value f_i), P_j is the average escape rate of all samples belonging to the f_j fitness level and L is the number of possible fitness levels. Being the mean value of a set of probabilities, the aep will be 0 when the instance is hard and higher (up to 1) otherwise. The calculation of the aep requires the analysis of the neighbourhood of each solution in order to identify how many individuals have a equal or better fitness than the original individual. We analyse, therefore, the evolvability of the solutions which have been selected (with probability equal to 0.2) for the local search. Since the calculation of the neighbourhood of each solution corresponds to the first step of the local search, no significant additional cost is required to compute the aep.

4.2 Dispersion metric

The analysis of the distribution of the solutions within the landscape can be sometimes used to understand more about the difficulty that a "jump" between fitness levels requires and to gain some information on the global structure of the landscape. In this context, the dispersion metric (dm) (Lunacek and Whitley 2006) is a technique to obtain information about the global structure of the landscape, by measuring the dispersion of good solutions. Ideally, if good solutions are very close, we might have a single funnel structure. If, on the contrary, solutions get more distant when their fitness improves, the landscape might be more like a multi-funnel structure. The analysis can be described as follows:

1. A sample S of solutions is taken from the search space;
2. the best S_{best} solutions are selected from S (using a threshold value);
3. the average pairwise distances in S ($\overline{d}(S)$) and in $S_{best}(\overline{d}(S_{best}))$ are calculated using the CARP diversity measure shown in Fig. 1;
4. the dm is obtained as the difference between $\overline{d}(S_{best})$ and $\overline{d}(S)$.

The calculation of the pairwise distance between all the individuals of the sample is already performed during the diversity-based stochastic ranking of MAENS* by using the distance measure shown in Fig. 1, and therefore requires no additional cost. Thus, the dm can computed on the set of all the *popsize*offset* individuals created during each generation of MAENS*. Finally, it is possible to rely on the ranking performed by the diversity-based stochastic ranking operator and choose these solutions as the subset of the best ones.

4.3 Average neutrality ratio and Δ-fitness

Neutrality is the study of the width, distribution and frequency of neutral structures within a landscape (e.g., plateaus, ridges). A set of several neutrality measures was defined in Vanneschi et al. (2006). Among these, we select the following two:

1. average neutrality ratio ($\overline{r}$): can be obtained by averaging the neutrality ratio (e.g., the number of solutions with equal fitness) of each individual with respect to its neighbourhood;
2. average Δ-fitness of neutral network ($\Delta(\overline{f})$): can be defined as the average fitness gain after one mutation step of each individual belonging to a neutral network.

In the same fashion as in the case of the aep, the computational effort of this technique can be absorbed by the generation of the neighbourhood of the initial solution during the local search.

5 Online learning

The AOS model followed by MAENS* is that of the multi-armed bandit approach, where the UCB1 (Auer et al. 2002) algorithm is used to balance the exploration and exploitation of the crossover operators and the Page-Hinckley (Hinkley 1971) test is used to detect when a different operator has become the most suitable.

In this work, we propose the adoption of a different model. The abrupt and scarcely predictable changes of the most suitable operator which might happen during the search show many similarities to the notion of concept drift (Schlimmer and Granger 1986; Minku et al. 2010) in machine learning.

Thus, in such a context, we might adopt an online learning algorithm capable of (a) predicting a reward for each operator using the online fitness landscape analysis measures and (b) tracking the changes of the environment, relying only on a limited number of training instances. We can define more formally the learning problem in the following way. At a given generation of the EA, we compute the FLA metrics (fla_1, fla_2, fla_3, fla_4) and the reward of each operator ($RW(op_i)$). Tuples (fla_1, fla_2, fla_3, fla_4, $RW(op_i)$) are then used as training examples for the online learning algorithm, where (fla_1, fla_2, fla_3, fla_4) are the input features and ($RW(op_i)$) is the target output.

We employ the dynamic weighted majority (DWM) (Kolter and Maloof 2003) algorithm as our online learning algorithm, which has proved to be one of the most effective techniques in the task of tracking concept drifts. The DWM algorithm can be described as follows. A set of learners (called experts) are used to classify the incoming instances $\{\overrightarrow{x}, y\}$, where $\overrightarrow{x}$ is the vector of n input features and y is the output feature. Each expert e_j has its own weight w_j, and operates a classification λ of the instance. The global prediction is identified as the prediction with the largest sum of weights. All the experts which have failed to classify correctly the instance have their weights reduced by a β factor. Moreover, for every p instances, all the experts with a weight below a certain threshold θ, are deleted and a new expert is created if the global prediction is wrong.

5.1 DWM for regression tasks

As the DWM algorithm was originally conceived for classification it is necessary to adapt and modify some of its mechanism for the regression task of predicting the reward of a given operator based on the FLA techniques. A comparison between the revised DWM algorithm for the regression task (rDWM) and the original DWM itself is given in Fig. 2. The modifications introduced are:

1. The global prediction σ_i is obtained by calculating the weighted average of all predictions (line 10);
2. we consider a prediction correct if its difference from the output feature is less than a threshold τ (lines 5–6);
3. a new expert is created if the difference between the global prediction and the output feature is less than a t factor (lines 17, 18);
4. we introduce a window containing the last n instances wTS, which is used to train the new experts upon creation (line 2).

6 MAENS*-II

The revised version of the algorithm adopting the rDWM as an AOS mechanism, named MAENS*-II, is shown in Fig. 3,

(a) Original DWM.

```
1  for (each instance {x⃗_i, y_i}) do
2      for (each expert e^j) do
3          λ^j = classify(e^j, x⃗_i);
4          if (|λ_i^j ≠ y_i) then
5              w_j = β * w_j;
6          end
7      end
8      normalize weights;
9      σ_i = select class with largest sum of weights;
10     if (p mod i = 0) then
11         for (each expert e^j) do
12             if (w_j < θ) then
13                 delete expert;
14             end
15         end
16         if (σ_i ≠ y_i) then
17             create new expert;
18         end
19     end
20     for (each expert e^j) do
21         train(e^j, x⃗_i);
22     end
23 end
```

(b) DWM for the regression task.

```
1  for (each instance {x⃗_i, y_i}) do
2      update wTS(x⃗_i);
3      for (each expert e^j) do
4          λ^j = predict(e^j, x⃗_i);
5          if (|λ_i^j − y_i| > tau) then
6              w_j = β * w_j;
7          end
8      end
9      normalize weights;
10     σ_i = global prediction
11     if (p mod i = 0) then
12         for (each expert e^j) do
13             if (w_j < θ) then
14                 delete expert;
15             end
16         end
17         if (|σ_i − y_i| > t) then
18             create new expert
                 and train using wTS;
19         end
20     end
21     for (each expert e^j) do
22         train(e^j, x⃗_i);
23     end
24 end
```

Fig. 2 Pseudocode of DWM (*left side*) and DWM for the regression task (*right side*). The novelties introduced in the latter version, discussed in Sect. 5.1, are highlighted with a *gray background color*

along with the original MAENS* algorithm. Further information about MAENS* can be found in Consoli and Yao (2014). A set of four (one for each crossover operator) rDWM instances are created upon initialization of the algorithm (line 2). During each generation, one new training example is created for each rDWM instance by using the current set of FLA metrics as input features, and the reward associated to the operator as the output feature (lines 10, 13, 14) obtained with a given credit reward strategy. The set of four rDWM instances are then used to predict the reward of each operator (line 4). Finally, an operator selection rule is adopted to choose the operators to use during each generation.

Three different versions of the MAENS*-II algorithm were implemented employing the two different techniques for the operator selection rule introduced in Sect. 3.2 as well as the two different credit assignment mechanism presented in Sect. 3.1. All the experiments were performed using the *weka* (Hall et al. 2009) implementation of REPTrees as base learners. Table 1 summarizes a list of the different versions of the algorithm and a description of their components. It is worth noting that the combination of the DBR strategy and the instantaneous reward was not considered, as the strategy of measuring the reward of the crossover offspring and the use of only one operator during each generation would

lead to the its exclusive use for the whole execution of the algorithm.

6.1 Improvements on local search efficiency

One of the most effective features of MAENS (Tang et al. 2009) is its local search, which, however, has a high computational cost—the algorithm spends around 95 % of its runtime performing this operation. Although the proposed modifications to the original MAENS algorithm, as explained in Sect. 4, cause no significant increase of the runtime of the algorithm, a fast implementation of MAENS local search is introduced, which helped reducing effectively the runtime without incurring into extra memory consumption. The approach is similar to the one introduced in Zachariadis and Kiranoudis (2010) for the vehicle routing problem, but without relying on the use of memory.

The approach can be summarized by the following points:

1. Every individual a in the neighbourhood of a solution x is represented as a *move M*, where M stores the information relative to the move operator op_i such that $op_i(x) = a$, the tasks involved in the move, and the variations in terms

(a) MAENS* pseudocode

1 initialize a population *pop* of *popsize* individuals;

2 **while** *(termination condition not met)* **do**

3   choose the crossover operator op_i using *dMAB*

4   generate a population pop_x of *popsize∗offset* individuals, choosing the parents from $pop \cup pop_x$;

5   generate $pop_{ls}(i)$ for each individual $pop_x(i)$ with probability = 0.2;

6   **if** *($pop_{ls}(i)$ is better than $pop_x(i)$)* **then**

7   overwrite $pop_x i$;

8   **end**

9   use d-Stochastic Ranking and overwrite *pop*;

10   use the PR measure to calculate the reward R_i for each op_i;

11 **end**

(b) MAENS*-II pseudocode

1 initialize a population *pop* of *popsize* individuals;

2 initialize four $rDWM_i$ instances

3 initialize four rewards rw_i

4 **while** *(termination condition not met)* **do**

5   choose the crossover operator op_i with largest rw_i

6   generate a population pop_x of *popsize∗offset* individuals, choosing the parents from $pop \cup pop_x$;

7   generate $pop_{ls}(i)$ for each individual $pop_x(i)$ with probability = 0.2;

8   **if** *($pop_{ls}(i)$ is better than $pop_x(i)$)* **then**

9   overwrite $pop_x i$;

10   **end**

11   calculate $Y^t = \{aep, \overline{r}, \Delta(\overline{f}), dm\}$

12   use d-Stochastic Ranking and overwrite *pop*;

13   **for** *each op_i* **do**

14         $out_i = CreditAssignment(op_i)$

15         $rw_i = rDWM_i(Y^t, out_i)$

16   **end**

17 **end**

Fig. 3 Pseudocode of MAENS* (*right side*) and MAENS*-II (*left side*). The novelties introduced in the latter version, discussed in Sect. 6, are highlighted with a *gray background color*

Table 1 List of algorithms

Combination	Operator selection rule	Reward measure
a	Concurrent strategy	Diversity-based reward
b	Concurrent strategy	Proportional reward
c	Instantaneous reward	Proportional reward

Each row represents a different combination of one operator selection rules and one reward measure strategies

of fitness and violation of the constraints w.r.t. the values of the initial solution.

2. The set of moves representing the whole neighbourhood is split into $V = R * R$ subsets, where R is the number of routes of the initial solution. Each subset contains the moves relative to the move operators applied to the tasks belonging to the routes R_i and R_j.

3. During the first iteration of the local search, the whole neighbourhood of moves is produced. A storage array of size V is kept to store the best solution of each subset.

4. The best move in the neighbourhood is identified with a computational time of $O(M)$. If the best move belongs to the subset relative to the routes R_i and R_j, the positions

in the storage relative to the combination of either routes are set to null.

5. In the following iterations, the local search produces only the moves involving either the route R_i or R_j or both. The positions of the storage relative to such moves are consequently updated.

After the first iteration, the number of subsets to be evaluated is therefore decreased from R^2 to $2R + 1$, resulting in a significant reduction in terms of size of the neighbourhood that is necessary to evaluate during each local search iteration. It is worth mentioning that the use of the *move* notation itself reduces the cost of evaluating the fitness and the violation of one individual from $O(n)$ to $O(k)$, where n is the number of tasks and k is equal either to 7 or 8 (depending on the move operator considered).

7 Experimental studies

A set of experiments were designed to understand the behaviour of MAENS*-II. As a first step, an oracle based on the Proportional Reward was implemented with the purpose of

Table 2 Characteristics of the instances of Groups A (top part) and B (bottom part)

Instance	$|V|$	$|R|$	$|E|$	BK	Instance	$|V|$	$|R|$	$|E|$	BK
C1	69	79	98	1590	e3-C	77	87	98	10,163
C10	60	55	82	2190	e4-A	77	98	98	6408
C11	83	94	118	1725	e4-B	77	98	98	8884
C15	97	107	140	1765	e4-C	77	98	98	11,427
C18	93	121	133	2315	S1-B	140	75	190	6384
C5	56	65	79	2410	S2-A	140	147	190	9824
C6	38	51	55	855	S2-B	140	147	190	12,968
C9	76	97	117	1775	s2-C	140	147	190	16,353
D1	69	79	98	725	s3-A	140	159	190	10,143
D11	83	94	118	920	s3-B	140	159	190	13,616
D21	60	76	84	695	s3-C	140	159	190	17,100
D23	78	92	109	715	s4-A	140	190	190	12,143
D7	54	52	70	735	s4-B	140	190	190	16,093
D8	66	63	88	615	s4-C	140	190	190	20,375
E1	73	85	105	1855	F1	73	85	105	1065
E11	80	94	113	1810	F11	80	94	113	1015
E12	74	67	103	1580	F12	74	67	103	900
E15	85	107	126	1555	F14	53	55	72	1025
E19	77	66	103	1400	F19	77	66	103	685
E21	57	72	82	1700	F24	97	86	142	975
E23	93	89	130	1395	F4	70	77	99	930
E5	68	61	94	2130	F7	73	50	94	1080
E9	93	103	141	2160	F9	93	103	141	1145
e1-B	77	51	98	4498	val4D	41	69	69	526
e2-B	77	72	98	6305	val5D	34	65	65	573
e3-B	77	87	98	7704	val8C	30	63	63	518
					val10D	50	97	97	525
EGL-G1-A	255	347	375	970,495	EGL-G2-A	255	375	375	1,061,103
EGL-G1-B	255	347	375	1,085,097	EGL-G2-B	255	375	375	1,173,286
EGL-G1-C	255	347	375	1,201,030	EGL-G2-C	255	375	375	1,295,036
EGL-G1-D	255	347	375	1,325,317	EGL-G2-D	255	375	375	1,430,267
EGL-G1-E	255	347	375	1,461,469	EGL-G2-E	255	375	375	1,557,159

For each instance the table provides informations such as the number of vertices of the graph ($|V|$), the number of required edges or tasks ($|R|$), the number of edges of the graph ($|E|$) and the best known solution in literature (BK)

analysing a set of CARP instances in order to obtain optimal crossover operator selection rates and to analyze them. The oracle can be briefly described as follows. Four different populations are obtained during each generation by using each crossover operator. All the individuals of the four generations are merged into a single population which is sorted using the MAENS* ranking operator. The proportional reward mechanism is therefore used to assess the best operator. The results achieved by the oracle show that the predictions operated by the dMAB are not optimal, as better results can be achieved. Besides, the results of the oracle should be considered "optimal" only when the proportional reward strategy is considered, because they might not nec-

essarily be optimal when in presence of a set of multiple measures, as in the case of MAENS*-II, or when a different credit assignment strategy is considered.

Two different datasets are considered for the experiments. The first one, named Group A, is composed of instances taken from the known benchmark test sets egl (Eglese 1994), Beullen's *C, D, E, F* (Beullens et al. 2003) and val (Benavent et al. 1992). The second group (Group B) corresponds to the large scale CARP instances of the dataset EGL-G (Brandão and Eglese 2008). The characteristics relative to each instance, in terms of number of vertices, number of edges, number of required edges and best fitness value found in literature are included in Table 2.

Table 3 Parameters of the MAENS*-II algorithms

Name	Description	Value
MAENS* parameters		
psize	Population size	30
ubtrial	Maximum attempts to generate a solution	50
opsize	Size of the offspring during each generation	6*psize
Parent selection	Crossover parent selection strategy	Random selection
P_{ls}	Probability of performing the local search	0.2
pMS	Routes selected during MergeSplit	2
G_{max}	Maximum generations	500
SR_{r1}	Probability of sorting solutions using diversity	0.25
SR_{r2}	Probability of sorting solutions using fitness	0.70
MAENS*-II parameters		
p	Expert removal period	5
β	Decrease factor for expert weights	0.75
τ	Expert weight reduction threshold	0.05
θ	Threshold for expert removal	0.05
t	Threshold for expert creation	0.10
ψ	Control parameter for concurrent strategy	0.002

In the upper part of the table, we report the set of parameters and the respective values that are shared with the MAENS* algorithm. In the bottom part of the table, the new parameters introduced for the MAENS*-II algorithm

Table 4 Statistics relative to the performances of the four crossover operators for datasets A and B

	GSBX	GRX	PBX	SPBX
Dataset A				
Fitness	18	14	14	12
Violation	17	12	13	16
Diversity	17	16	11	16
Parent distance	16	12	13	15
Dataset B				
Fitness	3	3	3	1
Violation	3	6	1	0
Diversity	4	3	0	3
Parent distance	0	3	4	3

The table shows the number of instances for which each of the four crossover operators (GSBX, GRX, PBX, SPBX) achieved the worst results in terms of fitness, violation, average pairwise diversity and distance from parents. The values are obtained averaging the statistics of four different populations of size 10,000 generated using each crossover respectively from the same parent population of 10,000 individuals

We provide an example of a solution for the *D07* problem instance produced by one of the variants of the MAENS*-II algorithm in Fig. 4, to clarify what kind of results this algorithm produces.

The set of parameters adopted in all the MAENS*-II algorithm variants, included in Table 3, were identified by a series of test-and-trial attempts and might not correspond to the most optimal choice. With regards to the parameters that are common with the other algorithms (MAENS and MAENS*) we adopted the same set of values in order to exclude different results due to different parameter configurations. These parameters can be identified in Table 3. All the final results were obtained by averaging the output of 30 independent runs.

7.1 Single operator scenario

In order to understand the improvement achievable by MAENS*-II, the algorithm was executed on the two benchmark sets considering each of the four available crossover operators. The results of such experiments for Group A are included in Table 5. For each single operator MAENS* version, the results show the average fitness over 30 independent runs, the standard deviation and the fitness of the best individual found. The last column, named *best*, shows the results of what an "optimal" adaptive operator selection would achieve (picking the best results out of the four achieved). In the second to last row (named #), the table provides the number of instances with statistically different results according to the results of a Wilcoxon rank-sum test with Holm–Bonferroni correction at the significance level of 0.05. The row at the bottom (named *W*) shows the number of comparison won against the other algorithms. A Wilcoxon rank-sum test was performed on the results achieved on every instance by each pair of algorithms, with Holm–Bonferroni correction to deal with the multiple comparisons. The results across all the problem instances were then compared using the Wilcoxon signed-rank test. Each problem instance with comparable results was treated as paired results and therefore omitted from the test. The results of such test, subject to the Holm–Bonferroni correction, are included at the bottom row (*pBest*) of Table 5.

For Group A, it is possible to notice how there is a great number of instances for which the four versions achieve statistically different results. The only exception is represented by the comparison between the *PBX*- and the *SPBX*-based versions, for which there is a limited number of statistically different instances (7); in all the other cases, the statistically different instances are at least 24. The GSBX-based operator seem to be the one performing the worst, losing the comparison to most of the instances, while the other three operators achieve the best results in a similar number of times. The statistical difference between the results of the GSBX version and the other three versions is also confirmed by the results

Table 5 Experimental results on Group A using alternatively each crossover operator

inst	GSBX			GRX			PBX			SPBX			best
	avg	std	best	avg	std	best	avg	std	best	avg	std	best	avg
C01	4175.33	26.80	4150	4159.00	14.80	4150	4151.67	4.53	4150	4153.17	7.13	4150	4151.67
C05	5373.75	15.29	5365	5366.00	2.00	5365	5366.50	2.29	5365	5367.50	2.50	5365	5366.00
C06	2545.75	4.82	2535	2537.50	4.61	2535	2541.00	4.90	2535	2540.33	4.99	2535	2537.50
C09	5274.92	19.27	5260	5281.17	19.57	5260	5262.83	9.97	5260	5263.33	9.52	5260	5262.83
C10	4709.42	16.26	4700	4709.00	12.21	4700	4712.33	10.14	4700	4703.33	7.45	4700	4703.33
C11	4648.58	16.33	4640	4643.17	3.29	4640	4641.33	2.21	4640	4641.50	3.20	4630	4641.33
C15	4964.42	16.43	4940	4946.83	10.76	4940	4946.50	5.19	4940	4946.33	4.27	4940	4946.33
C18	5639.83	9.83	5620	5642.17	6.28	5620	5635.00	8.37	5620	5640.17	9.26	5620	5635.00
D01	3225.00	9.04	3215	3235.00	0.00	3235	3230.83	5.01	3215	3229.67	6.45	3215	3225.00
D07	3115.83	2.76	3115	3115.33	1.80	3115	3115.00	0.00	3115	3115.67	2.49	3115	3115.00
D08	3052.83	4.12	3045	3058.00	10.38	3045	3045.67	2.49	3045	3047.67	4.42	3045	3045.67
D11	3761.08	3.17	3760	3760.33	3.14	3755	3760.83	2.27	3755	3760.17	3.76	3745	3760.17
D21	3058.33	3.25	3050	3056.67	2.98	3050	3059.83	2.41	3050	3060.00	2.24	3055	3056.67
D23	3171.17	11.74	3140	3158.17	8.51	3145	3187.17	10.30	3155	3177.50	10.47	3145	3158.17
E01	4916.83	5.84	4910	4910.33	1.25	4910	4912.00	3.32	4910	4912.67	4.23	4910	4910.33
E05	4623.33	21.67	4585	4623.33	27.34	4585	4607.33	19.09	4585	4608.67	18.39	4585	4607.33
E09	5855.33	25.00	5820	5838.83	23.55	5810	5836.50	19.88	5815	5832.67	17.83	5810	5832.67
E11	4697.25	24.91	4660	4671.67	4.35	4670	4675.00	11.25	4670	4678.33	12.67	4670	4671.67
E12	4228.50	17.28	4190	4209.33	19.09	4190	4200.67	11.95	4190	4201.83	9.79	4195	4200.67
E15	4220.67	7.04	4205	4215.00	6.06	4210	4221.50	5.19	4210	4220.00	6.19	4210	4215.00
E19	3244.17	2.76	3235	3239.33	4.96	3235	3244.67	1.80	3235	3243.83	3.08	3235	3239.33
E21	3733.50	2.29	3730	3731.33	2.21	3730	3734.50	1.50	3730	3734.00	2.00	3730	3731.33
E23	3718.83	5.87	3715	3715.17	1.57	3710	3715.33	1.80	3710	3717.33	2.81	3710	3715.17
egl-e1-B	4512.47	12.04	4498	4504.87	10.94	4498	4503.03	8.78	4498	4501.77	8.41	4498	4501.77
egl-e2-B	6328.65	11.75	6317	6321.93	8.68	6317	6322.60	6.03	6317	6327.10	10.19	6317	6321.93
egl-e3-B	7792.07	15.71	7775	7780.90	9.45	7775	7784.97	8.83	7777	7782.57	4.96	7777	7780.90
egl-e3-C	10328.18	19.82	10292	10324.73	16.07	10305	10315.87	17.93	10292	10310.67	16.24	10292	10310.67
egl-e4-A	6464.97	5.39	6444	6464.23	4.26	6461	6463.47	3.00	6456	6463.90	1.83	6461	6463.47
egl-e4-B	9021.28	17.37	8988	9059.70	25.38	8988	9024.40	15.16	8998	9013.10	14.09	8988	9013.10
egl-e4-C	12032.60	1047.53	11559	11593.13	22.87	11554	11586.40	18.91	11539	11584.97	25.66	11543	11584.97
egl-s1-B	6415.70	21.28	6388	6405.50	19.41	6388	6393.17	2.48	6388	6399.43	13.95	6388	6393.17
egl-s2-A	9942.62	26.54	9895	9949.67	24.62	9890	9929.37	23.69	9889	9939.50	27.44	9889	9929.37
egl-s2-B	13201.76	35.16	13144	13244.63	90.51	13137	13163.57	30.84	13103	13181.60	29.65	13122	13163.57
egl-s2-C	16500.12	42.51	16430	16480.80	39.54	16430	16456.77	17.46	16430	16462.37	27.46	16430	16456.77
egl-s3-A	10298.59	31.43	10221	10305.10	45.56	10242	10284.60	25.31	10233	10300.73	25.47	10253	10284.60
egl-s3-B	13847.47	60.89	13713	13906.90	50.92	13771	13792.63	40.81	13713	13815.87	61.71	13707	13792.63
egl-s3-C	17317.86	38.59	17209	17290.47	41.79	17197	17292.90	39.24	17242	17287.70	37.00	17221	17287.70
egl-s4-A	12409.36	41.76	12296	12438.10	33.13	12389	12367.40	38.27	12315	12399.70	34.21	12316	12367.40
egl-s4-B	16448.97	43.34	16316	16499.40	45.08	16430	16384.27	43.71	16292	16405.10	40.22	16329	16384.27
egl-s4-C	25200.21	2151.47	20781	22201.00	143.44	21792	20801.63	103.60	20601	20796.70	112.85	20584	20796.70
F01	4046.81	3.04	4040	4047.00	3.32	4040	4047.50	3.10	4040	4046.67	3.50	4040	4046.67
F04	3499.47	5.10	3485	3500.17	3.76	3495	3498.83	5.43	3485	3500.50	5.06	3485	3498.83
F07	3347.73	33.33	3335	3338.33	17.95	3335	3338.33	17.95	3335	3355.00	40.00	3335	3338.33
F09	4750.71	9.95	4730	4743.00	7.48	4730	4751.67	12.20	4730	4753.83	10.70	4740	4743.00
F11	3850.38	13.65	3835	3845.83	11.91	3835	3839.83	6.89	3835	3844.83	10.99	3835	3839.83

Table 5 continued

inst	GSBX			GRX			PBX			SPBX			best
	avg	std	best	avg	std	best	avg	std	best	avg	std	best	avg
F12	3416.14	28.91	3395	3450.33	21.21	3395	3423.67	22.10	3395	3410.17	16.71	3395	3410.17
F14	3339.29	11.03	3330	3349.33	18.25	3330	3349.00	6.88	3340	3359.33	14.13	3330	3339.29
F19	2553.97	9.38	2525	2531.17	10.46	2525	2565.33	9.21	2545	2564.83	12.88	2525	2531.17
F24	3234.38	11.30	3215	3230.17	13.01	3210	3245.50	7.89	3225	3244.17	6.20	3220	3230.17
val10D	531.47	1.79	528	532.73	1.26	530	530.73	1.18	528	530.50	1.52	528	530.50
val4D	532.03	3.01	530	531.63	2.44	530	530.27	0.85	530	530.00	0.00	530	530.00
val5D	583.37	1.83	579	586.80	3.53	578	581.87	2.38	575	583.60	2.42	579	581.87
val8C	524.37	1.96	521	526.80	1.08	525	524.50	1.84	521	525.00	1.83	521	524.37

	GRX	PBX	SPBX	GSBX	PBX	SPBX	GSBX	GRX	SPBX	GSBX	GRX	PBX
#	26	31	24	26	26	27	31	26	7	24	27	7
W	8	6	5	18	8	13	25	18	3	19	14	4
p	0.126	0.0004	0.0021	0.126	0.1738	0.3524	0.0004	0.1738	–	0.0021	0.3524	–
pBest	<0.001			0.002			0.005			0.007		

The first column shows the instance name (*inst*). For each operator (one among GSBX, GRX, PBX, spxb) the table includes the average fitness of the best solution (*avg*), the standard deviation (*std*), the best solution (*best*). Last column (*best*) shows the best *avg* result among the four crossover operators. In the rows at the bottom, the number of comparisons (#) against every operator with statistically different results according to the Wilcoxon rank-sum test with Holm–Bonferroni correction at the 0.05 significance level. The following row (*W*) shows the number of comparisons where the algorithm achieves a better average fitness. In the last rows (*p* and pBest), the *p* values relative to the Wilcoxon signed-rank test with Holm–Bonferroni correction at the 0.05 significance level between each single-based version and against the column of the best results

of the Wilcoxon signed-rank test. None of the four single operator-based versions of MAENS* algorithm is able to perform as good as the "optimal" results (in the *best* column), as testified by the results of the Wilcoxon signed-rank test included in the *pBest* row in Table 5.

The results of the comparison for the CARP instances belonging to the Group B are included in Table 6. The table shows the results of the four different versions of the algorithm, based on the use of one of the four crossover operators available. Analogously to the previous, the table presents the average fitness (*best*), the standard deviation (*std*) and the best result found (*best*) for each algorithm. The results show how the best results are achieved always by the *PBX*- and *SPBX*-based versions of the algorithm (providing better results on all statistically different instances against the other two versions). The *GRX*-based version, in contrast, is the one that performs the worst (losing the comparison five times out of six against the *GSBX*-based version and on all the statistically different instances for the other two versions).

In both datasets, SPBX and PBX operators appear to be the operators whose usage leads to the best results. This can be explained by the fact that such operators can introduce a fair amount of diversity in the offspring as one or more routes are built from scratch. On the other hand, they maintain the good traits of the parents copying the routes that are not affected by the recombination. The GSBX operator, in contrast, might not introduce much diversity in the off-

spring as the new routes are a combination of the subroutes of the parents. Therefore, despite being the least disruptive operator, on the long term it produces a minor contribution than SPBX and PBX. The GRX operator, on the other hand, has a larger disruptive capacity as only the best routes are preserved in the offspring. In the context of large instances with a great number of routes as in the case of dataset B, therefore, this operator might introduce an excessive level of exploration and consequently perform worse than the others.

A further experiment was conducted to analyse the behaviour of the four crossover operators. A population of 10,000 solutions was generated using the initialization operator. Each operator was then used to generate a population of 10,000 solutions, using a random parent selection mechanism. Table 4 reports the number of instances for which the operators achieved the worst results in terms of fitness, violation, average pairwise distance of the offspring population and average distance from the parents. This experiment has been repeated for both datasets. The results show that in dataset A the operator GSBX achieves the worst results in the largest number of instances for each of the characteristics analysed. This is coherent with the results achieved by the four evolutionary algorithms. On the other end, for dataset B, GRX is the worst algorithm for both fitness and violation and the second worst for both the diversity measures, which reflects the behaviour of the algorithm. It is however worth specifying that these results refer to the behaviour of the oper-

Table 6 Experimental results on Group B using alternatively each crossover operator

inst	GSBX			GRX		
	avg	std	best	avg	std	best
EGL-G1-A	983,408.50	3866.28	974,054	979,095.97	4718.81	968,747
EGL-G1-B	1,091,041.60	5539.83	1,081,857	1,094,405.97	7413.40	1,079,590
EGL-G1-C	1,213,921.77	5924.27	1,198,728	1,212,862.87	7906.85	1,198,393
EGL-G1-D	1,337,645.60	6620.72	1,322,885	1,336,158.93	6527.20	1,326,125
EGL-G1-E	1,473,433.10	5394.11	1,461,472	1,472,399.20	6338.61	1,461,155
EGL-G2-A	1,107,790.40	5057.72	1,099,946	1,138,050.43	10215.79	1,110,914
EGL-G2-B	1,225,440.43	5982.44	1,214,762	1,255,569.68	16865.83	1,224,099
EGL-G2-C	1,359,221.30	3798.77	1,349,981	1,404,364.32	10734.40	1,369,046
EGL-G2-D	1,497,934.97	6544.95	1,486,595	1,563,310.77	5689.30	1,552,126
EGL-G2-E	1,641,472.67	8022.86	1,626,564	1,713,877.53	8561.59	1,687,159
	GRX	PBX	SPBX	GSBX	PBX	SPBX
#	6	7	7	6	8	7
W	5	0	0	1	0	0
	PBX			SPBX		
	avg	std	best	avg	std	best
EGL-G1-A	980,746.30	5681.29	970,911	**978,411.33**	4308.05	969,682
EGL-G1-B	1,087,682.37	5542.29	1,074,857	**1,087,255.93**	4986.75	1,079,899
EGL-G1-C	**1,208,196.23**	5138.15	1,198,557	1,210,928.20	5972.77	1,202,072
EGL-G1-D	**1,330,286.83**	6554.80	1,321,271	1,333,503.57	6436.41	1,324,605
EGL-G1-E	**1,462,940.17**	5293.51	1,452,158	1,467,270.13	5003.86	1,458,893
EGL-G2-A	**1,104,884.20**	3252.86	1,099,756	1,105,976.47	3662.89	1,098,458
EGL-G2-B	1,221,379.87	3586.68	1,213,622	**1,220,895.90**	4180.51	1,212,440
EGL-G2-C	**1,351,635.77**	5294.39	1,343,015	1,354,111.13	5087.04	1,343,399
EGL-G2-D	**1,490,662.50**	4435.48	1,484,014	1,492,414.43	4495.21	1,484,208
EGL-G2-E	1,635,578.87	4851.51	1,623,322	**1,634,917.43**	5199.54	1,623,417
	GSBX	GRX	SPBX	GSBX	GRX	PBX
#	7	8	1	7	7	1
W	7	8	1	7	7	0

The results of the four versions of the algorithm are split in two different rows. The first column shows the instance name (*inst*). For each operator (one among GSBX, GRX, PBX, spxb) the table includes the average fitness of the best solution (*avg*), the standard deviation (*std*), the best solution (*best*). In the rows at the bottom, the number of comparisons (#) against every operator with statistically different results according to the Wilcoxon rank-sum test with Holm–Bonferroni correction at the 0.05 significance level. The following row shows the number of comparisons where the algorithm achieves a better average fitness
Bold results are the best results achieved among the four operators

ators with a population of low quality solutions (as they are generated through the use of the initialization operator) and might not necessarily reflect the behaviour of the crossover operators during the most advanced phases of the search.

7.2 Operator selection rules and reward measures: a comparison

The performance of the algorithm using different operator selection rules and reward measures is shown in Tables 8 and 9, respectively, for the groups A and B. We include the results of the three combinations introduced in Table 1, along with the optimal result considering the best performance of the single operator versions (in the last column, named *best*). In Table 8, the results of the statistical tests show how the three versions of the algorithm achieve statistically different results only on a limited subset of the instances (at most 7 between MAENS*-IIa and MAENS*-IIc). The versions achieving the best results appear to be the ones adopting the concurrent strategy as an OSR (MAENS*-IIa and MAENS*-IIb). The two versions achieve extremely similar results (differing only in three instances), while the version using the instantaneous

Table 7 Experimental results on the instances of Group A relative to *MAENS**, *MAENS*-IIrw*, the *oracle*, and MAENS* with random selection

inst	MAENS*			MAENS*IIrw			Oracle			Random		
	avg	std	best	avg	std	best	avg	std	best	avg	std	best
C01	4161.67	19.38	4150	4158.67	13.16	4150	4155.33	13.03	4150	4164.67	18.48	4150
C05	5366	10.95	5365	5378.33	18.36	5365	5365	0	5365	5366.17	2.11	5365
C06	2542	25.1	2535	2545.17	3.98	2535	2536.67	3.73	2535	2541.33	4.82	2535
C09	5270	91.65	5260	5280.33	20.41	5260	5269	15.08	5260	5272.83	17.01	5260
C10	4702.17	6.54	4700	4707.33	11.53	4700	4700.67	3.59	4700	4702.17	6.54	4700
C11	4641.33	3.14	4630	4657	27.37	4640	4640.17	2.41	4630	4642.33	2.49	4640
C15	4946.17	7.38	4940	4964.17	15.76	4940	4947	9.27	4940	4946.17	3.80	4940
C18	5638.67	7.74	5620	5642.17	6.91	5625	5636.17	7.82	5625	5638.33	10.03	5620
D01	3232.83	4.02	3215	3224.83	8.99	3215	3229.5	6.87	3215	3231.17	5.87	3215
D07	3115	0	3115	3116.33	3.4	3115	3115	0	3115	3115	0	3115
D08	3045.67	2.49	3045	3052	4.58	3045	3045.67	2.49	3045	3046.33	3.40	3045
D11	3761.5	3.91	3760	3762.67	6.42	3745	3759.67	4.99	3745	3760.17	2.41	3750
D21	3059.83	5.24	3050	3063.67	11.47	3055	3055.17	3.98	3050	3058.17	3.02	3050
D23	3164.83	12.28	3135	3167.83	12.23	3140	3153.17	8.51	3135	3165.83	14.55	3140
E01	4911.17	2.11	4910	4916	6.11	4910	4910.5	1.5	4910	4910.83	2.27	4910
E05	4606.67	22.34	4585	4621.5	21.57	4585	4612	24.17	4585	4605.83	22.10	4585
E09	5837	21.16	5815	5851.33	25.26	5815	5835.83	21.26	5815	5840.17	22.49	5810
E11	4677	13.52	4655	4698	25.68	4670	4673.83	7.71	4665	4678.83	12.23	4670
E12	4202.33	13.15	4180	4226	17.63	4195	4204.5	11.5	4190	4203.67	11.32	4190
E15	4217.5	6.68	4205	4223.67	5.91	4210	4214.5	6.24	4205	4217.67	6.02	4210
E19	3242.67	4.23	3235	3244.67	1.8	3235	3238.33	4.71	3235	3242	4.58	3235
E21	3733	2.45	3730	3732.67	2.49	3730	3730.67	1.7	3730	3733.17	2.41	3730
E23	3715.5	1.5	3715	3720.5	7.34	3715	3714	2	3710	3716	2.71	3710
e1-B	4501.2	8.33	4498	4509.17	11.68	4498	4502.6	8.5	4498	4500.80	7.44	4498
e2-B	6323.67	9.58	6317	6329.83	13.35	6317	6320.37	6.36	6317	6324.17	8.96	6317
e3-B	7780.43	5.91	7775	7790.47	11.23	7777	7783.93	11.61	7775	7778.43	3.22	7775
e3-C	10,317.6	18.45	10,292	10,323.6	20.38	10,292	10,316.63	18.86	10,292	10,313.07	15.52	10,292
e4-A	6462.5	3.04	6450	6464.07	5.39	6446	6462.77	2.58	6456	6462.13	5.20	6446
e4-B	9022.5	16.39	8988	9023.47	16.23	8992	9011.2	11.79	8993	9032.60	15.95	8999
e4-C	11,592.53	32.82	11,538	11,602.8	31.64	11,550	11,610.13	41.31	11,554	11,593.50	21.02	11,555
s1-B	6399.9	16.38	6388	6407.3	19.35	6388	6399.7	14.5	6388	6399.87	15.42	6388
s2-A	9931.63	26.62	9889	9943.43	32.78	9889	9928.37	27.01	9885	9933.97	29.35	9889
s2-B	13,179.07	26.11	13,124	13,217.13	44.41	13,159	13,179.2	29.61	13,124	13,181.57	32.72	13,125
s2-C	16,510.1	43.05	16,430	16,516.03	46.02	16,430	16,498	41.64	16,433	16,512.27	39.21	16,442
s3-A	10,282.63	29.41	10,221	10,293.87	29.07	10,242	10,276.5	26.39	10,221	10,288.63	28.17	10,243
s3-B	13,820.13	57.75	13,736	13,874.37	59.29	13,736	13,823.37	60.51	13,750	13,818.93	62.17	13,714
s3-C	17,289.73	42.75	17,220	17,325.9	46.56	17,237	17,296.1	33.42	17,249	17,286.87	32.29	17,215
s4-A	12,400.87	47.91	12,283	12,403.37	47.36	12,316	12,382.93	41.71	12,304	12,407.07	31.77	12,305
s4-B	16,421.17	50.46	16,325	16,454.3	42.73	16,351	16,414.67	47.18	16,344	16,435.90	33.33	16,334
s4-C	21,047.97	174.66	20,758	21,065.8	166.32	20,702	21,117.7	327.1	20,745	20,955.50	181.04	20,611
F01	4046.83	2.73	4040	4046.43	2.54	4040	4044.5	3.73	4040	4046.17	3.34	4040
F04	3498.67	3.64	3485	3499.67	5.31	3485	3496.17	3.8	3485	3499	4.16	3485
F07	3335	0	3335	3345	30	3335	3345	30	3335	3341.67	24.94	3335
F09	4746	11.79	4730	4750.53	12.34	4730	4742	8.12	4730	4746.67	10.83	4730

Table 7 continued

inst	MAENS*			MAENS*IIrw			Oracle			Random		
	avg	std	best	avg	std	best	avg	std	best	avg	std	best
F11	3850	11.11	3835	3846.97	13.56	3835	3841	6.88	3835	3851.50	12.12	3835
F12	3410	23.42	3395	3425.7	32.32	3395	3402.33	13.09	3395	3412.17	16.82	3395
F14	3342.33	12.23	3330	3340.83	13.17	3330	3338	13.52	3330	3344.33	12.16	3330
F19	2535.17	9.35	2525	2537.67	8.73	2525	2526.67	3.73	2525	2544.67	12.78	2525
F24	3234.33	8.63	3215	3232	9.36	3215	3225.5	11.28	3210	3239.33	9.64	3220
val10D	530.6	1.23	528	532.13	1.65	530	529.9	1.08	528	531.03	1.35	528
val4D	530.13	0.72	530	530.77	2.04	530	530.23	0.76	530	530.37	1.17	530
val5D	583.13	2.06	579	583.97	3.42	577	581.93	2.69	577	583.57	2.29	579
val8C	524.63	1.76	521	524.23	2.49	521	523.37	1.76	521	524.93	1.77	521
	MII*a	MII*b	MII*c	M*IIc			M*IIa		M*IIb	M*IIa	M*IIb	M*IIc
#	4	2	4	36			19		18	8	8	4
W	0	0	2	3			16		18	0	0	0

The first column shows the instance name (*inst*). For each version of the algorithm tested the table includes the average fitness of the best solution (*avg*), the standard deviation (*std*), the best solution (*best*). In the rows at the bottom, the number of comparisons against every operator with statistically different results according to the Wilcoxon rank-sum test with Holm–Bonferroni correction at the 0.05 significance level. The following row shows the number of comparisons where the algorithm achieves a better average fitness

reward (MAENS*-IIc) differs from the other two respectively in seven and five instances, and loses the comparison in the majority of the cases. In contrast, MAENS*-IIc is the variant that differs the least w.r.t. the best results achieved by the single operator versions of the algorithm (only six statistically different instances, while the other two versions differ in eight ad nine instances).

Although such results show small differences between the performances of the algorithms when adopting one OSR rather than the other, it is possible to see how the concurrent strategy appears to perform slightly better. This might be explained by several factors. First, the use of more than one crossover operator might introduce higher diversity in the whole offspring population. Secondly, the capacity of monitoring and verifying the performance of all the crossover operators might be important to detect changes in the environment. With regards to the reward measure adopted, the two approaches achieved similar results. This could be interpreted by similar importance of the requirements that the two measures try to satisfy (diversity and survival ability of the offspring). The balance might be different when tackling larger CARP instances, as in the case of those in Group B, where the exploration ability of the operator might have a bigger impact on the performance of the algorithm.

The results achieved by MAENS*-IIa and MAENS*-IIb on Group B are included in Table 9. The two algorithms show a comparable result on nine instances out of ten, with the only statistically different result according to the Wilcoxon rank-sum test being that of the instance EGL-G2-C, with a *p*-value of 0.0004. The similarity of the results achieved by the two

different versions of the algorithm in both datasets can be explained by the fact that the use of the FLA metrics makes the algorithm more robust with respect to the reward measure considered. Further experiments with more aggressive credit assignment strategies might reveal more differences between the adoption of the two different reward measures. Finally, we provide a comparison with a version the algorithm selecting one crossover operator randomly during each generation. The results of such algorithm are included in Table 7, in the rightmost column named *random*. At the bottom it is possible to see the number of statistically different instances according to the Wilcoxon rank-sum test with a level of significance of 0.05 (line #) with respect to the three algorithms MAENS*-IIa, MAENS*-IIb and MAENS*-IIc, along with the number of times the random algorithm has won the comparison (line *W*). It is worth noting that the random algorithm achieves a fairly good performance, as it achieves statistically comparable results with the proposed techniques for most of the instances. This result could be interpreted as a probable sign of positive interaction between the crossover operators that have been considered in this case study.

7.3 Effectiveness of the FLA measures

An experiment was designed to understand whether the use of the online FLA techniques has a beneficial effect on both the optimization ability and the prediction capacity of the algorithm. Therefore, MAENS*-IIc was compared to MAENS*-rw, a version of the algorithm which only makes use of the Proportional Reward measure as an input feature of

Table 8 Experimental results on the instances of Group A for *MAENS*-IIa*, *MAENS*-IIb* and *MAENS*-IIc*

inst	MAENS*-IIa			MAENS*-IIb			MAENS*-IIc			best
	avg	std	best	avg	std	best	avg	std	best	avg
C01	4159.50	17.53	4150	4156.50	12.66	4150	4160.00	17.75	4150	4151.67
C05	5367.17	3.34	5365	5366.83	2.41	5365	5369.00	5.83	5365	5366.00
C06	2540.00	5.00	2535	2539.67	4.99	2535	2541.00	4.90	2535	2537.50
C09	5268.00	16.00	5260	5261.67	7.23	5260	5264.00	10.12	5260	5262.83
C10	4707.33	9.64	4700	4704.17	8.37	4700	4705.17	9.53	4700	4703.33
C11	4641.33	3.14	4630	4641.50	2.29	4640	4642.33	2.49	4640	4641.33
C15	4944.67	7.74	4940	4945.50	4.72	4940	4946.00	7.57	4940	4946.33
C18	5641.17	9.04	5620	5637.50	8.73	5620	5637.76	9.27	5620	5635.00
D01	3232.83	3.10	3225	3231.17	6.67	3215	3232.17	5.11	3215	3225.00
D07	3115.00	0.00	3115	3115.00	0.00	3115	3115.00	0.00	3115	3115.00
D08	3045.33	1.82	3045	3047.67	4.42	3045	3047.67	4.42	3045	3045.67
D11	3760.33	2.25	3755	3760.50	1.50	3760	3761.72	3.48	3755	3760.17
D21	3058.33	3.24	3050	3058.33	3.25	3050	3059.00	4.16	3050	3056.67
D23	3172.67	9.66	3150	3162.50	13.34	3135	3162.67	7.39	3150	3158.17
E01	4911.00	2.41	4910	4911.00	2.00	4910	4911.50	2.93	4910	4910.33
E05	4611.83	25.35	4585	4601.17	18.74	4585	4615.00	26.08	4585	4607.33
E09	5830.67	18.35	5810	5832.17	19.39	5810	5834.17	21.64	5810	5832.67
E11	4674.33	10.70	4670	4672.33	5.12	4670	4678.00	15.03	4660	4671.67
E12	4201.00	7.39	4195	4205.33	12.24	4195	4207.50	14.59	4180	4200.67
E15	4218.00	6.98	4210	4217.83	5.73	4205	4219.33	5.59	4210	4215.00
E19	3243.33	3.78	3235	3242.00	4.58	3235	3242.00	4.58	3235	3239.33
E21	3733.50	2.31	3730	3733.67	2.21	3730	3733.27	2.39	3730	3731.33
E23	3715.83	3.23	3710	3716.83	2.73	3715	3715.50	1.98	3710	3715.17
e1-B	4503.60	9.87	4498	4499.67	6.26	4498	4504.79	10.42	4498	4501.77
e2-B	6324.67	10.28	6317	6321.63	5.84	6317	6323.86	9.41	6317	6321.93
e3-B	7783.77	9.12	7775	7782.87	8.35	7777	7786.55	10.73	7777	7780.90
e3-C	10,312.23	15.45	10,292	10,314.80	20.03	10,292	10,318.31	19.15	10,292	10,310.67
e4-A	6463.87	3.30	6454	6463.07	2.02	6461	6463.83	5.07	6446	6463.47
e4-B	9029.27	16.82	9000	9026.63	16.17	9000	9021.10	17.84	8990	9013.10
e4-C	11,589.13	24.44	11,540	11,586.80	27.09	11,536	11,621.28	72.42	11,555	11,584.97
s1-B	6402.53	18.33	6388	6401.80	16.88	6388	6397.59	12.70	6388	6393.17
s2-A	9931.93	25.17	9889	9928.37	24.06	9889	9934.80	29.49	9881	9929.37
s2-B	13,171.97	24.27	13,122	13,170.97	31.06	13,107	13,171.41	29.10	13,123	13,163.57
s2-C	16,478.50	34.87	16,425	16,492.30	39.99	16,442	16,505.97	51.89	16,434	16,456.77
s3-A	10,282.67	32.08	10,221	10,288.47	28.39	10,221	10,290.67	25.78	10,251	10,284.60
s3-B	13,814.90	58.66	13,722	13,818.63	73.32	13,717	13,821.50	47.04	13,747	13,792.63
s3-C	17,287.27	37.04	17,205	17,288.43	31.12	17,223	17,309.87	37.46	17,221	17,287.70
s4-A	12,404.20	35.59	12,301	12,388.17	37.70	12,304	12,388.59	41.42	12,316	12,367.40
s4-B	16,399.90	50.38	16,305	16,427.13	51.61	16,278	16437.60	54.52	16,281	16,384.27
s4-C	20,847.80	134.55	20,603	20,912.60	266.38	20,565	21,037.24	223.95	20,648	20,796.70
F01	4047.00	2.79	4040	4045.50	3.25	4040	4047.59	3.32	4040	4046.67
F04	3498.67	3.45	3485	3498.17	4.74	3485	3499.00	4.16	3485	3498.83
F07	3348.33	30.45	3335	3345.00	30.00	3335	3338.34	17.95	3335	3338.33
F09	4749.50	12.21	4730	4746.67	10.43	4730	4748.45	8.48	4730	4743.00
F11	3843.67	11.25	3835	3843.17	11.58	3835	3847.07	12.35	3835	3839.83

Table 8 continued

inst	MAENS*-IIa			MAENS*-IIb			MAENS*-IIc			best
	avg	std	best	avg	std	best	avg	std	best	avg
F12	3404.67	11.93	3395	3408.00	17.45	3395	3416.83	26.94	3395	3410.17
F14	3342.50	9.62	3330	3343.33	16.35	3330	3339.50	11.86	3330	3339.29
F19	2541.67	9.59	2525	2533.17	6.39	2525	2532.50	9.64	2525	2531.17
F24	3235.33	10.33	3215	3232.50	9.46	3215	3233.83	10.38	3210	3230.17
val10D	530.77	0.94	528	530.70	1.39	528	531.13	1.26	529	530.50
val4D	530.73	2.25	530	530.13	0.43	530	530.53	1.65	530	530.00
val5D	582.67	2.67	577	582.97	2.07	579	581.93	2.83	577	581.87
val8C	524.40	2.12	521	524.73	1.57	522	524.73	2.00	521	524.37
	b	c	best	a	c	best	a	b	best	
#	3	7	9	3	5	8	7	5	6	
W	1	5	0	2	5	0	2	0	0	

The first column shows the instance name (*inst*). For each version of the algorithm tested the table includes the average fitness of the best solution (*avg*), the standard deviation (*std*), the best solution (*best*). In the rows at the bottom, the number of comparisons against every operator with statistically different results according to the Wilcoxon rank-sum test with Holm–Bonferroni correction at the 0.05 significance level. The following row shows the number of comparisons where the algorithm achieves a better average fitness

Table 9 Experimental results on the instances of Group B for MAENS*-IIa and MAENS*-IIb

inst	BK	MAENS*-IIa			MAENS*-IIb		
		avg	std	best	avg	std	best
EGL-G1-A	970,495	978,636.00	5267.70	964,014	978,127.07	5330.95	968,157
EGL-G1-B	1,085,097	1,086,113.80	4709.52	1,075,069	1,088,504.40	5597.17	1,076,011
EGL-G1-C	1,201,030	1,209,512.20	5983.41	1,197,057	1,208,264.80	6225.46	1,196,975
EGL-G1-D	1,325,317	1,331,918.77	5702.86	1,322,682	1,331,367.00	5963.31	1,318,679
EGL-G1-E	1,461,469	1,466,771.97	6458.51	1,451,314	1,465,321.17	4890.69	1,455,995
EGL-G2-A	1,061,103	1,107,519.20	3744.93	1,099,674	1,107,461.13	2864.67	1,101,083
EGL-G2-B	1,173,286	1,220,912.47	4687.58	1,213,516	1,220,423.67	4751.97	1,213,237
EGL-G2-C	1,295,036	1,356,660.60	4883.30	1,346,969	1,352,307.90	5182.32	1,338,497
EGL-G2-D	1,430,267	1,493,163.27	5173.56	1,482,470	1,494,645.93	5294.38	1,486,269
EGL-G2-E	1,557,159	1,635,756.30	5750.90	1,622,468	1,636,974.10	6235.37	1,626,530

The first column shows the instance name (*inst*). The second column shows the fitness of the best known (BK) solution for each instance (Martinelli et al. 2013). For each version of the algorithm tested the table includes the average fitness of the best solution (*avg*), the standard deviation (*std*), the best solution (*best*)

the learning algorithm, without considering the values provided by the FLA techniques. In this context, we are not interested in the results achieved by the algorithm but rather we want to verify that the results are significantly different or not and prove, as a consequence, a certain suitability of the rDWM algorithm to the presence of the FLA measures. The results of such algorithm are included in Table 7 in the column *MAENS*-IIrw*. A Wilcoxon ranked-sum test was performed against the results achieved by MAENS*-IIc. The two algorithms produced statistically different results on 36 instances out of 53. MAENS*-IIrw achieved better results only on three instances, losing the comparison on 33. A Wilcoxon signed-rank test was consequently applied

across the problem instances, which confirmed that the two algorithms produce significantly different results (respectively $W_{\text{stat}} = 26$ with $p < 0.05$ and $W_{\text{stat}} = 54.5$ sample size: 42). This can be interpreted as a signal that the rDWM is concretely affected by the FLA measures, which influence (in a beneficial way) the decisions made by the algorithm.

7.4 Comparison with the state-of-the-art

The second research question in the introduction of this paper focuses on the performance of the proposed approach with respect to the existing ones. Therefore, the MAENS*-II vari-

Table 10 Comparison with some state-of-the-art approaches for CARP

inst	MAENS*		MAENS-RDG		VNS		ILS-RVND	
	avg	best	avg	best	avg	best	avg	best
EGL-G1-A	977,754.4	968,897	1,007,223.0	1,000,575	10,007,393.0	997,055	1,014,930.9	1,004,864
EGL-G1-B	1,088,706.5	1,079,793	1,124,751.0	1,111,971	1,122,077.0	1,114,120	1,143,221.7	1,129,937
EGL-G1-C	1,209,058.8	1,195,902	1,251,718.0	1,243,779	1,253,789.0	1,243,808	1,270,100.3	1,262,888
EGL-G1-D	1,331,595.8	1,323,397	1,383,619.0	1,371,443	1,383,997.0	1,373,480	1,409,811.9	1,398,958
EGL-G1-E	1,469,455.4	1,449,542	1,524,393.0	1,512,584	1,525,994.0	1,517,772	1,556,138.5	1,543,804
EGL-G2-A	1,107,363.0	1,101,559	1,108,916.0	1,096,027	1,105,870.0	1,098,454	1,126,561.0	1,115,339
EGL-G2-B	1,223,132.3	1,213,769	1,222,183.0	1,213,617	1,220,012.0	1,211,759	1,237,741.8	1,226,645
EGL-G2-C	1,354,725.3	1,345,587	1,353,118.0	1,344,148	1,351,845.0	1,344,184	1,376,931.6	1,371,004
EGL-G2-D	1,495,089.7	1,486,646	1,489,723.0	1,481,181	1,489,500.0	1,481,045	1,520,794.3	1,509,990
EGL-G2-E	1,636,140.6	1,630,656	1,630,132.0	1,618,955	1,630,048.0	1,616,119	1,664,230.2	1,659,217

The first column shows the instance name (*inst*). For each algorithm the table includes the average fitness of the best solution (*avg*) and the best solution (*best*). The results are compared to those achieved by MAENS*-IIa and MAENS*-IIb included in Table 9. The algorithms considered are MAENS* (Consoli and Yao 2014), MAENS-RDG (Mei et al. 2014a), VND (Mei et al. 2014b) and ILS-RVND (Martinelli et al. 2013). No statistical test was carried out due to the partial availability of the results of the compared algorithms

ants that make use of the Proportional Reward (a and b) were tested against the oracle. All the three variants were also compared against the results achieved by their base algorithm MAENS*. The results achieved by the oracle and by the MAENS* algorithm for Group A are included in Table 7, in columns *MAENS** and *oracle*. In the bottom rows, the results of a Wilcoxon rank-sum test with Holm–Bonferroni correction at the 0.05 significance level show the number of instances with statistically different results. The results of the statistical test show how the number of statistically different results is small (4 for MAENS*-IIa and MAENS*-IIc and 2 for MAENS*-IIb). In these few instances, MAENS*-IIa and MAENS*-b perform better than MAENS*, while MAENS*-II wins the comparison in half of the instances (2 out of 4). The online learning system is therefore able to achieve results comparable to those achieved by the bandit solver.

The comparison with the oracle shows that MAENS*-IIa and MAENS*-IIb are able to achieve comparable results in most cases. In most of the instances with statistically different results, the oracle was able to perform better. It is worth noting that in a small number of instances the algorithm using the FLA measures was able to produce better results than the oracle. This is some evidence that, if the oracle represents a "lower bound" for the results that is possible to achieve using the proportional reward, the use of more than one measures (as in this case) can help the algorithm to achieve results beyond these bounds.

Finally, the results achieved by MAENS*-IIa and MAENS*-IIb, included in Table 9, are compared against four state-of-the-art algorithms, whose results are included in Table 10. We consider the results of MAENS* (Consoli and Yao 2014), of MAENS-RDG (Mei et al. 2014a) and VND (Mei et al. 2014b) and an algorithm combining iterate local search and variable neighbourhood descent (Martinelli et al. 2013).

It is possible to notice how MAENS*-IIa and MAENS*-IIb, as well as MAENS*, outperform all the other algorithms in terms of solution quality for the first five instances of Group B (Table 9). MAENS*-IIa, MAENS*-IIb and MAENS* produce a new best known solution for all of these instances, with MAENS*-IIa achieving the best ones on the first two instances (G1-A and G1-B), MAENS*-IIb on instances G1-C and G1-D and MAENS* finding the best one on the instance G1-E. In all these instances MAENS*-IIa and MAENS*-IIb achieve also the best average fitness in four cases. For the following five instances, the best results are achieved by either MAENS-RDG or VND. In all these cases, MAENS* is outperformed by both MAENS*-IIa and MAENS*-IIb. These results can be explained by the fact that their base algorithm, MAENS*, is already performing well for these instances. However, both variants MAENS*-IIa and MAENS*-IIb managed to outperform MAENS* in most of the instances. It is important to note that the runtime (not considered in this work) of these algorithms is not comparable to those of the decomposition-based approaches, which manage to find these results in a fraction of the time required by MAENS*-IIa and MAENS*-IIb.

The behaviour of the algorithms can be analyzed also in terms of the fitness distribution of its solutions. Figure 5 shows the box plot relative to three representative instances belonging to Group A (*egl-e4-C*, *egl-s1-B*, *egl-s2-B*) and one instance of Group B (*EGL-G2-A*). In the case of the *EGL-G1-A* instance, it is possible to notice how SPBX and GRX are the crossover operators whose usage leads to the distributions with the lowest median. The distribution of the three AOS considered in this case (MAENS*, MAENS*-

(a)

(b)   **(c)**   **(d)**   **(e)**
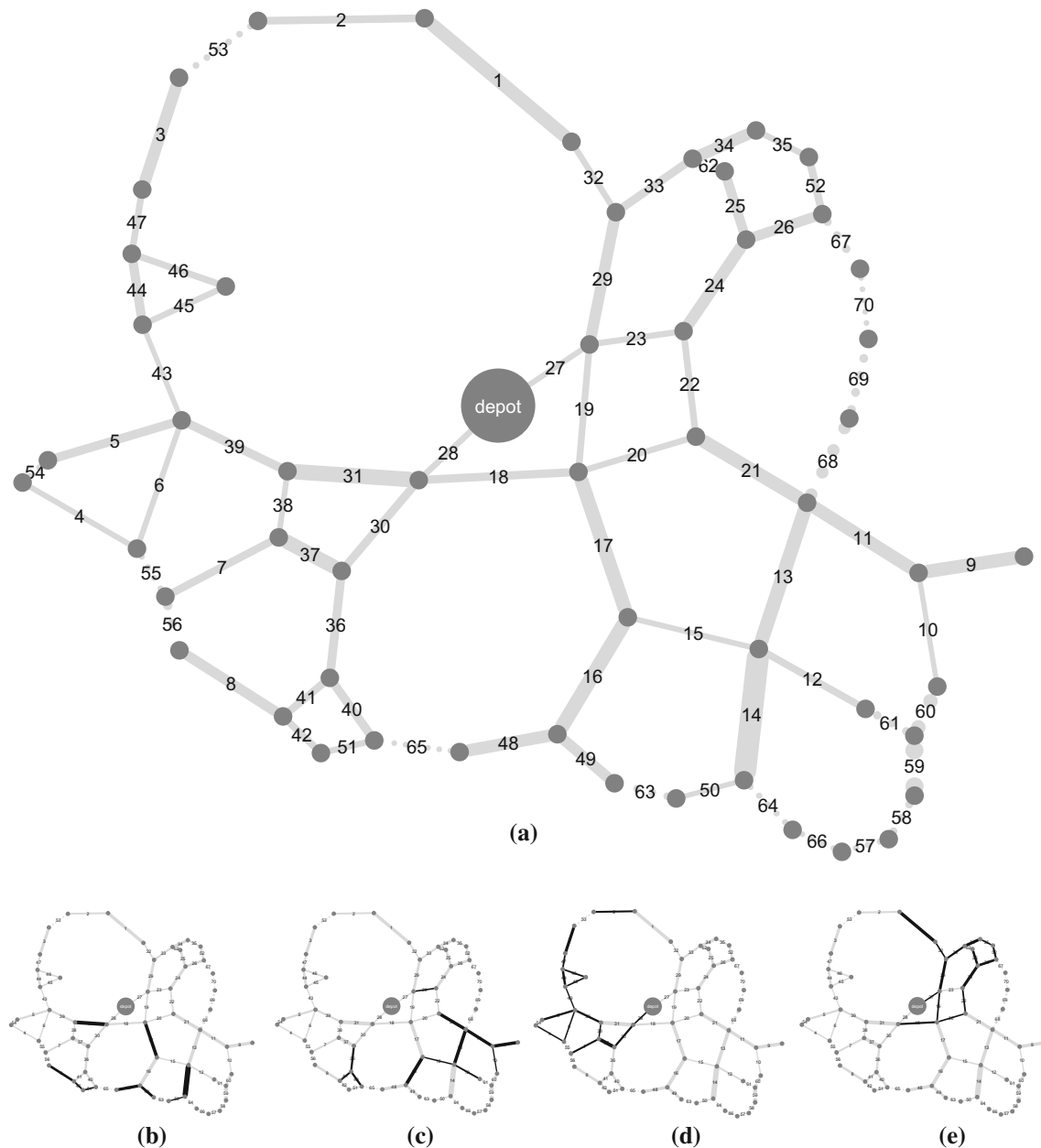
Fig. 4 The network relative to the *D07* instance (**a**). The cost of serving each edge is proportional to the *thickness of the line*. Non required edges can be identified by *dotted lines*. In **b–e** the four routes that compose a solution for this problem instance, generated by the MAENS*-IIa algorithm. The *edges* served during each route have been highlighted in *black*

IIa and MAENS*-IIb) are centered around the same median value, although MAENS*-IIa is capable of producing solutions of considerably better quality (bottom whisker) which translate into new minima for this instance. For the *egl-s1-B* instance (Fig. 5b), the behaviour of the algorithms is quite similar, as in most of the cases the distributions lie around the same median. When considering the results of the versions of the algorithm using each crossover operator, GSBX, GRX and SPBX show a much wider distribution of their results although in the first two cases a large number of solutions are equal to the median value, while PBX results are much less spread. The different AOS strategies achieve overall comparable results. This instance represents an example of non optimal behaviour as none of the AOS strategies considered has managed to match that of the best crossover operator (GRX).

For *egl-s2-b* (Fig. 5c), PBX is the operator that achieves the best results, while GRX performs the worst. MAENS*-IIb manages to achieve the same solution quality and similar median to PBX. This is also confirmed by the larger selection rate given to the PBX operator (Fig. 7b).

(a)

(b)

(c)

(d)
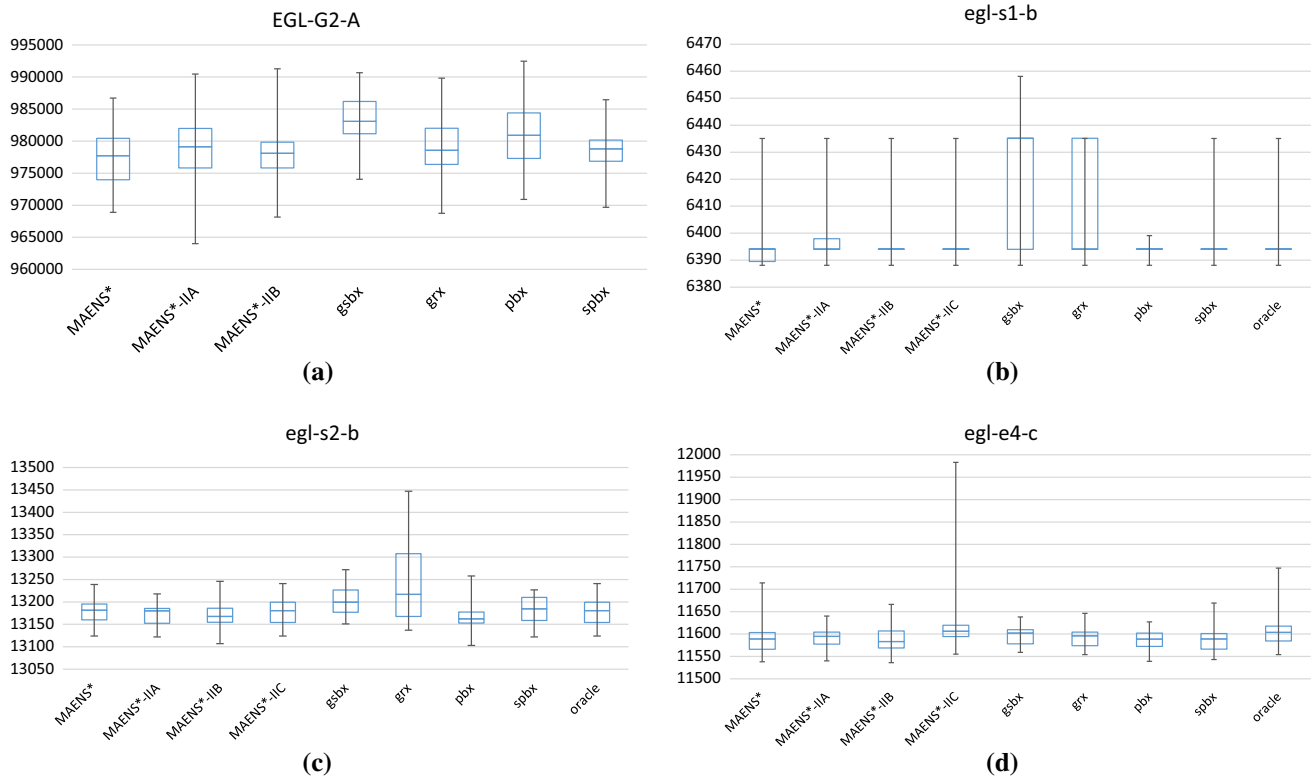
Fig. 5 *Boxplots* relative to the four instances *EGL-G2-A* (**a**), *egl-s1-b* (**b**), *egl-s2-b* (**c**), *egl-e4-c* (**d**). The *boxes* refer to the first quartile, median and third quartile. *Whiskers* show minimum and maximum values over a sample size of 30 best fitness values relative to the independent runs of each algorithm reported in Tables 5, 6, 7, 8, 9 and 10

In the case of *egl-e4-c* (Fig. 5d), PBX and SPBX distributions have a similar median and similar quartiles performing the best among the four crossover operators. Among the AOS strategies, MAENS*-IIb solutions are distributed around a similar median but more spread.

7.5 Prediction ability

To understand the behaviours of the algorithms, and to gain a deeper understanding of the selection mechanisms, we provide a comparison of the selection rates of the four different crossover operators, included in Figs. 6, 7 and 8. The plots refer to the selection rates relative to the instances *egl-s1-B*, *egl-s2-B* and *EGL-G2-A*. The *y*-axis in the figure refers to the selection rate (SR) of each crossover operators, where a SR of 0 means that the operator is not selected and a SR equal to 1 means that only that operator is selected. The *x*-axis corresponds to the average fitness of the population discretised into 50 intervals. We study, therefore, how the SR of the four operator changes while the search is carried out and the average fitness of the population decreases.

In the first instance, *egl-s1-B* (Fig. 6), it is possible to notice three phases in the oracle prediction (Fig. 6e). A first phase where the GRX operator is preferred over the others, an intermediate phase where the GRX and GSBX operators have nearly equal selection rates and a last phase characterized by a rise of the selection rate of the GRX operator which reaches 1 in the last moments of the search.

Both MAENS* (Fig. 6d) and MAENS*-IIc (Fig. 6c) award the GSBX operator with the highest selection rate for the whole search, missing the prediction of the change in the environment made by the oracle. It is possible to see, however, how MAENS*-IIc increases the selection rate of GSBX more rapidly than MAENS*.

The SR of both MAENS*-IIa (Fig. 6a) and MAENS*-IIb (Fig. 6b) show different changes during the search, proving that the CS is more successful in predicting such events. In particular, MAENS*-IIb acknowledges the operators GSBX and PBX as the most useful ones during the search. It is worth remembering that MAENS*-IIa, makes use of a different reward measure and, therefore, is not comparable to the prediction made by the oracle. In this case, MAENS*-IIa, after an initial epoch of dominance of the operator *GRX*, shows an alternance of moments where the three operators *GRX*, *PBX* and *SPBX* show the highest selection rates.

On the second instance (Fig. 7), the oracle identifies a change in the environment halfway through the search

(a)

(b)

(c)

(d)

(e)
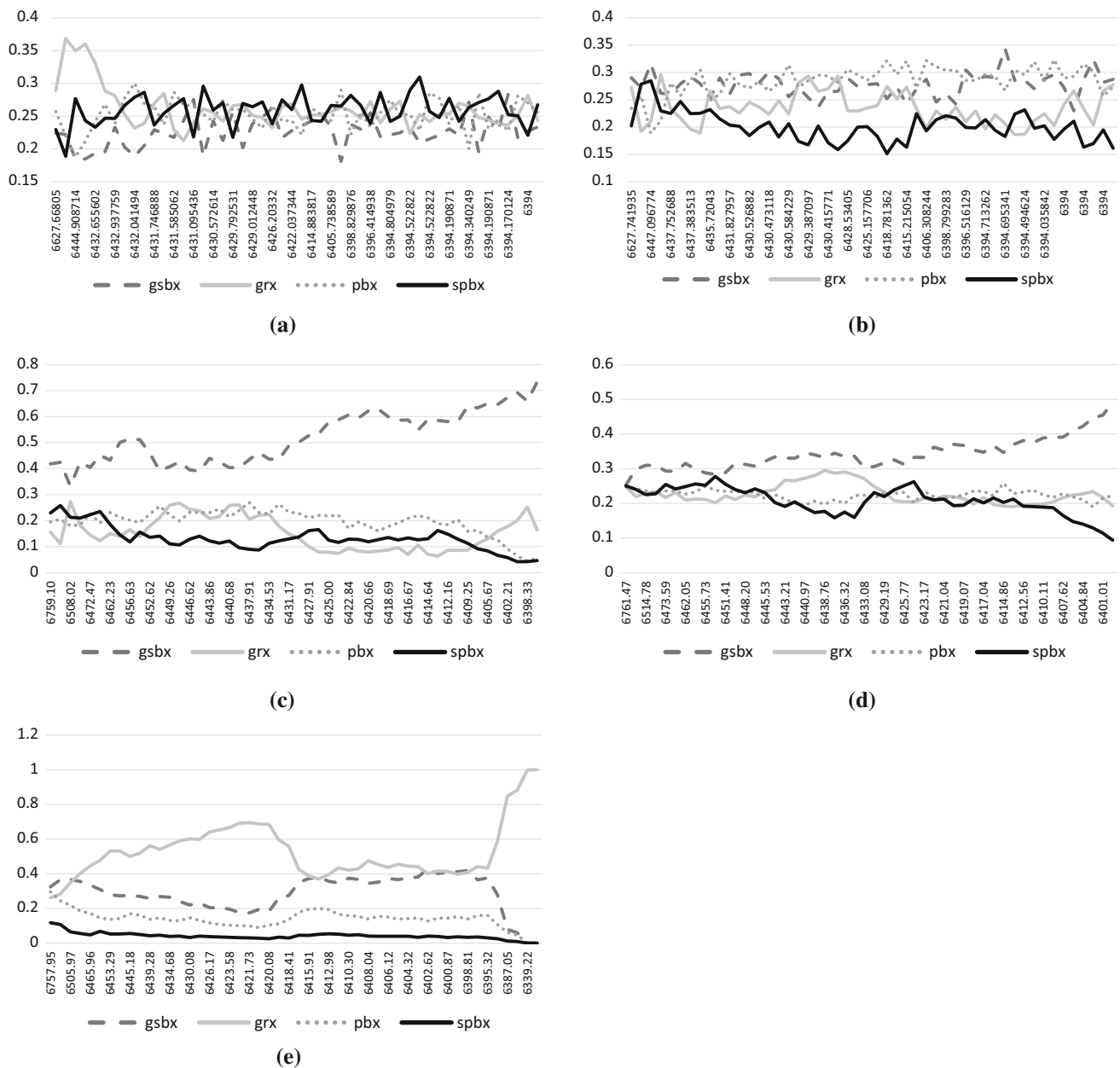
Fig. 6 Selection rates for the instance *egl-s1-b*. Each graph shows the selection rates of the four different crossover operators (*GSBX, GRX, PBX, SPBX*) when using respectively MAENS*-IIa (**a**), MAENS*-IIb (**b**), MAENS*-IIc (**c**), MAENS* (**d**) and the oracle (**e**)

(Fig. 7e). The concept drift is not detected by either MAENS* (Fig. 7d) or MAENS*-IIc (Fig. 7c), which, however shows an higher exploitation of the GSBX operator. MAENS*-IIb (Fig. 7b) identifies the operators *GSBX* and *PBX* as the most successful ones; even in this case the change detected by the operator is not detected. As for the previous instance, MAENS*-IIa (Fig. 7a) shows different moments where the three operators *GRX*, *PBX* and *SPBX* achieve the highest SR. The lowest SR for *GSBX* seems to indicate that this operator is probably the one that introduces the least diversity in the population.

For the large CARP instance EGL-G2-A, the behaviour of MAENS* (Fig. 8c) shows a predominance of operator *GSBX* over the other ones. MAENS*-IIb (Fig. 8b) shows a similar behaviour to that of MAENS*, identifying the *GSBX* operator as the one with the best performance during almost the whole search. Finally, MAENS*-IIa shows again an initial period of higher performance for the *GRX* operator, followed by an alternance of the *PBX* and *SPBX* operators (Fig. 8a). The occurrence of this initial period of higher performance for the *GRX* operator seems to suggest that this operator is introducing the highest diversity in the initial part

(a)

(b)

(c)

(d)

(e)
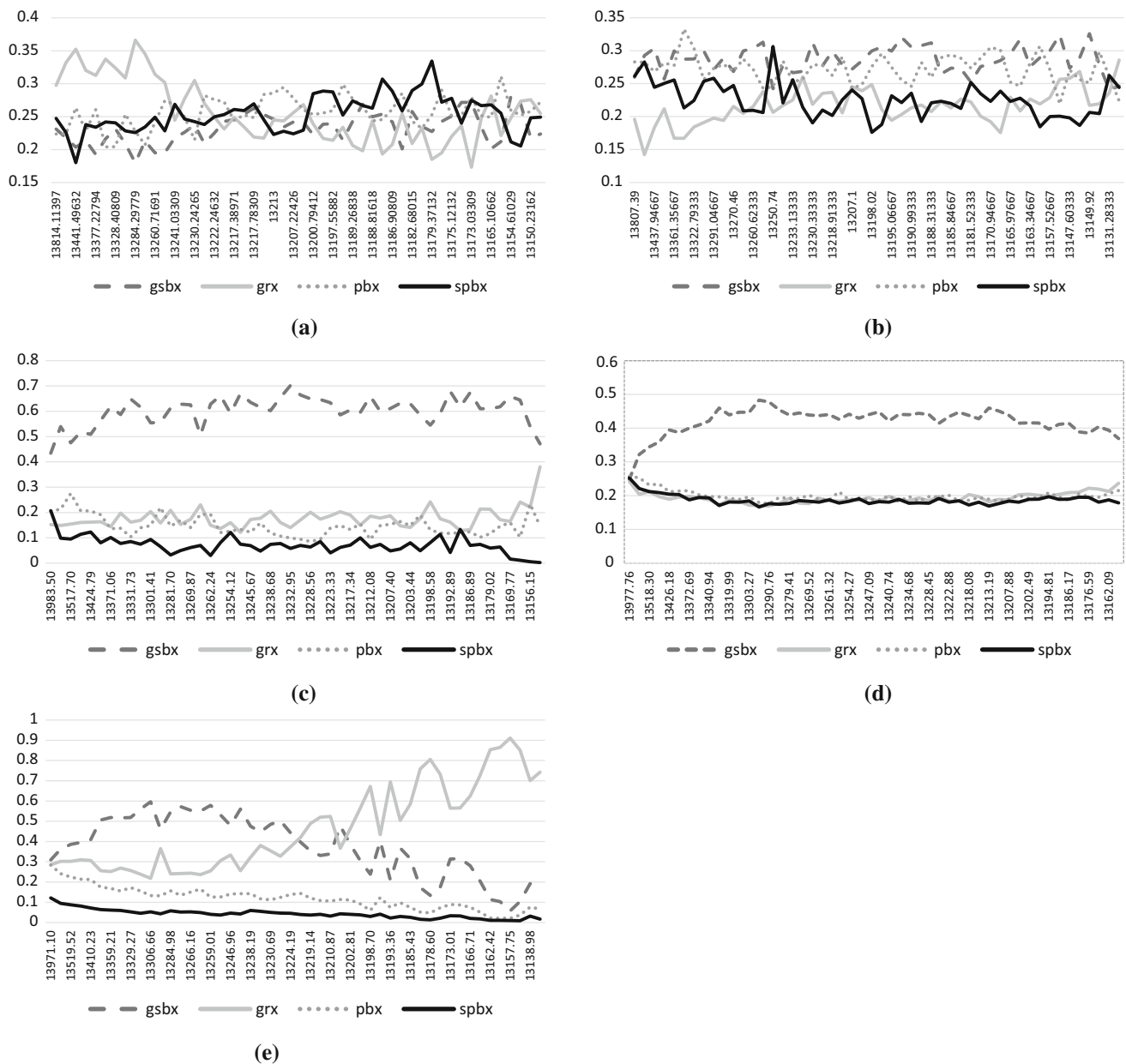
Fig. 7 Selection rates for the instance *egl-s2-b*. Each *graph* shows the selection rates of the four different crossover operators (*GSBX, GRX, PBX, SPBX*) when using respectively MAENS*-IIa (**a**), MAENS*-IIb (**b**), MAENS*-IIc (**c**), MAENS* (**d**) and the oracle (**e**)

of the search, when the solutions are not extremely good (Fig. 8a).

The results of these experiments show that failing to detect a change in the environment does not necessarily translate into a worst performance of the algorithm and vice versa. This is confirmed by the fact that the algorithms produce good results despite the different selection rates. The relationship between the prediction ability of the algorithms and their results is, therefore, quite complex. There are several factors that influence its behaviour and that should be considered in order to fully grasp this mechanism, such as the interaction between the different operators, the performance of the single operators and the variation of the selection rates.

8 Conclusions and future work

In this work, we proposed the adoption of a novel adaptive operator selection scheme to identify the optimal crossover operator online. We consider the use of two different reward measure strategies, the diversity-based reward (DBR) and the proportional reward (PR), as well as two different operator selection rule, namely the instantaneous reward (IR)

(a)

(b)

(c)
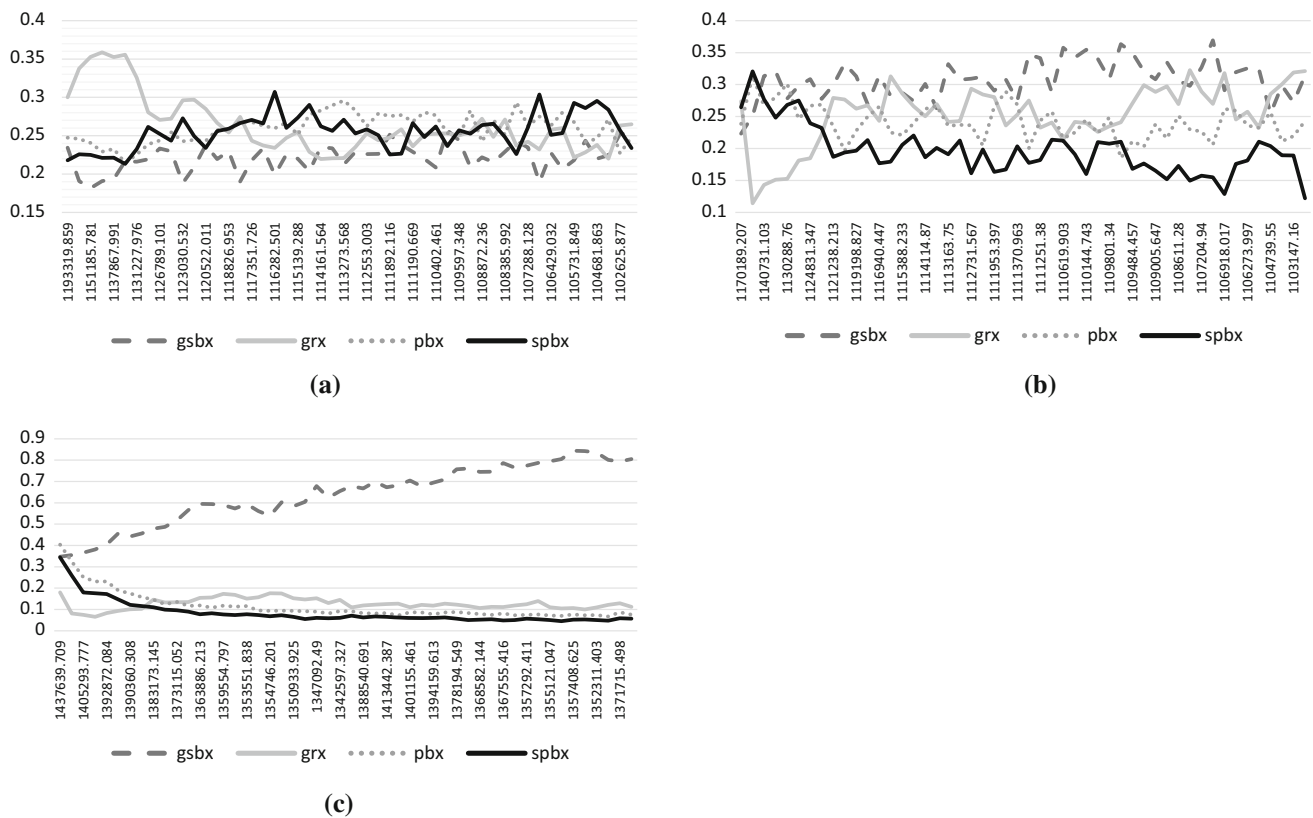
Fig. 8 Selection rates for the three instances *EGL-G1-A*. Each *row* shows the selection rates of the four different crossover operators (*GSBX, GRX, PBX, SPBX*) when using respectively MAENS*-IIa (**a**), MAENS*-IIb (**b**) and MAENS* (**c**). We do not include the results on EGL-G2-A for MAENS*-IIc as only versions MAENS*-IIa and MAENS*-IIb were tested on this dataset, as well as the oracle results, due to the extremely high computational cost required to perform this task

and a concurrent approach (CA). The AOS proposed combines a set of four fitness landscape analysis measures in conjunction with an online learning algorithm, to predict the most suitable crossover operator. We have chosen four FLA metrics to be used as inputs of our predictive model: accumulated escape probability, dispersion metric, average neutrality ratio and average delta fitness of neutral networks. These metrics have been chosen because (1) they can be computed without much increasing the computational effort and (2) they complement each other by capturing different features of the landscapes. Three versions of the MAENS* (Consoli and Yao 2014) algorithm were implemented and tested on two datasets of CARP instances. The results of such experiments were compared against those by state-of-the-art algorithms, and against an oracle. The results achieved by MAENS*-II show that this technique is able to compete with the state-of-the-art techniques and can, in some cases, exploit the multiple measures to outperform the state-of-the-art. In the dataset containing large CARP instances, MAENS*-II was able to outperform all the existing approaches in terms of average and best solution quality in half of the instances, and even discovered new lower bounds.

Our experiments seem to suggest a better performance of the concurrent strategy over the instantaneous reward, and a comparable performance of the two reward measure strategies.

This work leaves space for interesting directions that can be explored. First, the two reward measures might be combined to generate a novel measure that is able to predict better both the diversity and the survival ability of the offspring. Secondly, it would be interesting to test the behaviour of our algorithm when adopting an average or extreme reward strategy and the use of different base learners. Adaptive operator selection might be extended to different cases. In particular, fo MAENS, an AOS strategy can be adapted to choose among different parent selection strategies for the crossover operator, to analyze its impact on the offspring generation. Another direction is that of reducing the computational cost of MAENS*-II. Furthermore, due to the improved optimization ability provided by this approach, it would be interesting to test the use of MAENS*-II as the single objective routine for existing decomposition-based approaches. Finally, our technique might be adopted to improve the performance of evolutionary algorithms for other combinatorial optimization problems.

Compliance with ethical standards

Conflict of interest The authors declare no conflict of interest regarding the publication of this article.

References

Auer P, Cesa-Bianchi N, Fischer P (2002) Finite-time analysis of the multiarmed bandit problem. Mach Learn 47(2–3):235–256

Barbosa HJC, Sá AM (2000) On adaptive operator probabilities in real coded genetic algorithms. In: Workshop on advances and trends in artificial intelligence for problem solving – SCCC 2000 (2000)

Belluz J, Gaudesi M, Squillero G, Tonda A (2015) Operator selection using improved dynamic multi-armed bandit. In: Proceedings of the 2015 on genetic and evolutionary computation conference. ACM, New York, pp 1311–1317

Benavent E, Campos V, Corberán A, Mota E (1992) The capacitated arc routing problem: lower bounds. Networks 22(7):669–690

Beullens P, Muyldermans L, Cattrysse D, Van Oudheusden D (2003) A guided local search heuristic for the capacitated arc routing problem. Eur J Oper Res 147(3):629–643

Brandão J, Eglese R (2008) A deterministic tabu search algorithm for the capacitated arc routing problem. Comput Oper Res 35(4):1112–1126

Chen W, Wang Y, Yuan Y (2013) Combinatorial multi-armed bandit: general framework and applications. In: Proceedings of the 30th international conference on machine learning, pp 151–159

Consoli P, Yao X (2014) Diversity-driven selection of multiple crossover operators for the capacitated arc routing problem. In: Blum C, Ochoa G (eds) 14th European conference on evolutionary computation in combinatorial optimisation (EvoCOP'14), Granada. Revised selected papers, no. 12 in lecture notes in computer science. Springer, New York, pp 97–108

Consoli PA, Minku LL, Yao X (2014) Dynamic selection of evolutionary algorithm operators based on online learning and fitness landscape metrics. In: Simulated evolution and learning. Springer, New York, pp 359–370

DaCosta L, Fialho A, Schoenauer M, Sebag M (2008) Adaptive operator selection with dynamic multi-armed bandits. In: Proceedings of the 10th annual conference on genetic and evolutionary computation. ACM, New York, pp 913–920

Davis L (1989) Adapting operator probabilities in genetic algorithms. In: International conference on genetic algorithms'89, pp 61–69

Eglese RW (1994) Routeing winter gritting vehicles. Discret Appl Math 48(3):231–244

Eiben AE, Hinterding R, Michalewicz Z (1999) Parameter control in evolutionary algorithms. IEEE Trans Evol Comput 3(2):124–141

Eiben A, Horvath M, Kowalczyk W, Schut MC (2007) Reinforcement learning for online control of evolutionary algorithms. In: Engineering self-organising systems. Springer, New York, pp 151–160

Fialho Á, Da Costa L, Schoenauer M, Sebag M (2009) Dynamic multi-armed bandits and extreme value-based rewards for adaptive operator selection in evolutionary algorithms. In: Learning and intelligent optimization. Springer, New York, pp 176–190

Goldberg DE (1990) Probability matching, the magnitude of reinforcement, and classifier system bidding. Mach Learn 5(4):407–425

Golden BL, Wong RT (1981) Capacitated arc routing problems. Networks 11(3):305–315

Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The weka data mining software: an update. ACM SIGKDD Explor Newsl 11(1):10–18

Hinkley DV (1971) Inference about the change-point from cumulative sum tests. Biometrika 58(3):509–523

Julstrom BA (1995) What have you done for me lately? Adapting operator probabilities in a steady-state genetic algorithm. In: Proceedings of the 6th international conference on genetic algorithms, Pittsburgh

Karafotias G, Hoogendoorn M, Eiben A (2015) Evaluating reward definitions for parameter control. In: Applications of evolutionary computation. Springer, New York, pp 667–680

Kim M, McKay RIB, Kim DK, Nguyen XH (2012) Evolutionary operator self-adaptation with diverse operators. In: Genetic programming. Springer, New York, pp 230–241

Kolter JZ, Maloof M (2003) Dynamic weighted majority: a new ensemble method for tracking concept drift. In: Third IEEE international conference on data mining (ICDM'03). IEEE, pp 123–130

Lardeux F, Saubion F, Hao JK (2006) Gasat: a genetic local search algorithm for the satisfiability problem. Evol Comput 14(2):223–253

Lu G, Li J, Yao X (2011) Fitness-probability cloud and a measure of problem hardness for evolutionary algorithms. In: Evolutionary computation in combinatorial optimization. Springer, New York, pp 108–117

Lunacek M, Whitley D (2006) The dispersion metric and the CMA evolution strategy. In: Proceedings of the 8th annual conference on genetic and evolutionary computation. ACM, New York, pp 477–484

Martinelli R, Poggi M, Subramanian A (2013) Improved bounds for large scale capacitated arc routing problem. Comput Oper Res 40(8):2145–2160

Maturana J, Saubion F (2008) A compass to guide genetic algorithms. In: Parallel problem solving from nature-PPSN X. Springer, New York, pp 256–265

Mei Y, Li X, Yao X (2014a) Cooperative coevolution with route distance grouping for large-scale capacitated arc routing problems. IEEE Trans Evol Comput 18(3):435–449

Mei Y, Li X, Yao X (2014b) Variable neighborhood decomposition for large scale capacitated arc routing problem. In: 2014 IEEE congress on evolutionary computation (CEC). IEEE, pp 1313–1320

Merz P (2004) Advanced fitness landscape analysis and the performance of memetic algorithms. Evol Comput 12(3):303–325

Minku LL, White AP, Yao X (2010) The impact of diversity on online ensemble learning in the presence of concept drift. IEEE Trans Knowl Data Eng 22(5):730–742

Runarsson TP, Yao X (2000) Stochastic ranking for constrained evolutionary optimization. IEEE Trans Evol Comput 4(3):284–294

Sakurai Y, Takada K, Kawabe T, Tsuruta S (2010) A method to control parameters of evolutionary algorithms by using reinforcement learning. In: 2010 sixth international conference on signal-image technology and internet-based systems (SITIS). IEEE, pp 74–79

Schlimmer JC, Granger RH (1986) Beyond incremental processing: tracking concept drift. In: AAAI, pp 502–507

Soria Alcaraz JA, Ochoa G, Carpio M, Puga H (2014) Evolvability metrics in adaptive operator selection. In: Proceedings of the 2014

conference on genetic and evolutionary computation. ACM, New York, pp 1327–1334

Tang K, Mei Y, Yao X (2009) Memetic algorithm with extended neighborhood search for capacitated arc routing problems. IEEE Trans Evol Comput 13(5):1151–1166

Thierens D (2005) An adaptive pursuit strategy for allocating operator probabilities. In: Proceedings of the 2005 conference on genetic and evolutionary computation. ACM, New York, pp 1539–1546

Vanneschi L, Pirola Y, Collard P (2006) A quantitative study of neutrality in GP boolean landscapes. In: Proceedings of the 8th annual conference on genetic and evolutionary computation. ACM, New York, pp 895–902

Zachariadis EE, Kiranoudis CT (2010) A strategy for reducing the computational complexity of local search-based methods for the vehicle routing problem. Comput Oper Res 37(12):2089–2105