On the Use of Survival Selection Methods for Evolutionary Diversity Optimisation

Adel Nikfarjam
Optimisation and Logistics
School of Computer Science and Information Technology
Adelaide University

Jakob Bossek
Department of Computer Science
Machine Learning and Optimisation Group (MALEO)
Paderborn University

Aneta Neumann
Optimisation and Logistics
School of Computer Science and Information Technology
Adelaide University

Frank Neumann
Optimisation and Logistics
School of Computer Science and Information Technology
Adelaide University

***Abstract*—Generating a diverse set of high quality solutions for an optimisation problem has been studied extensively in recent years by the evolutionary computation community. A paradigm that has received increasing attention is evolutionary diversity optimisation (EDO), where the goal is to maximise the diversity of a solution set subject to quality constraints. Since the contribution of each solution to the diversity of the population depends on other solutions and can change dramatically if several solutions in the population are modified simultaneously, most EDO approaches generate a single new solution per generation and discard the solution with the least contribution to diversity, ensuring a steady increase in population diversity over successive generations until convergence. In this study, we aim to answer two questions: (1) Is generating multiple solutions in each generation beneficial for EDO? (2) How can this be achieved efficiently, given that conventional survival selection methods do not work well in EDO due to the dependency of a solution's contribution to diversity on other solutions?**

I. Introduction

The standard goal in single-objective optimisation is to find one single solution that is optimal or of high quality. Over the last decade, there has been a significant rise in the number of studies seeking a diverse set of high quality solutions for a given single-objective optimisation problem. Having a diverse set of high quality solutions can be valuable for a wide range of reasons, such as providing information about the solution space, offering decision-makers a variety of diverse alternatives, and improving robustness against imperfect modelling and dynamic changes of the problem.

Traditionally, evolutionary computation (EC) research considers diversity as a means to escape local optima and prevent premature convergence, or to explore niches in the fitness landscape, which is often referred to as multimodal optimisation. Niching has been studied extensively in the literature; interested readers are referred to [1] for a review of niching methods. In recent years, quality diversity (QD) and evolutionary diversity optimisation (EDO), have emerged in the EC literature.

QD explores a predefined behavioural space and seeks the best-performing solutions in different regions of this space. The framework was first proposed in [2], while an algorithm to depict the distribution of high-performing solutions in a behavioural space was introduced simultaneously in [3]. Later, the framework was formalised and named in [4], [5]. QD has been studied in various domains such as combinatorial optimisation problems [6], [7], improvement of health outcome through time-use plans [8], conditional search-space problems [9]; however, most QD studies focus on robotics and video games [10], [11].

On the other hand, EDO maximises the structural diversity of a fixed-size set of solutions, subject to solution quality. This means that the objective function of the given optimisation problem is treated as a constraint, while the diversity measure plays the role of the fitness function. Although EDO was first introduced in a continuous domain [12], it has been studied extensively in the context of combinatorial optimisation in recent years. Several studies have incorporated EDO to generate diverse sets of travelling salesperson problem (TSP) instances and images [13], [14], [15], [16], [17]. There have also been several papers employing EDO to generate diverse solutions for different combinatorial problems. EDO has been utilised to generate diverse set of solutions for many optimisation problems: travelling salesperson problem (TSP) [18], [19], [20], Quadratic Assignment Problem [21], the minimum spanning tree problem [22], monotone sub-modular functions [23], the travelling thief problem [24], the detection and concealment of communication networks in large settings [25], and the boolean satisfiability problem [26]. In most of these studies, a vanilla $(\mu + 1)$ EA has been utilised to maximise diversity, where at most one solution was replaced in each generation. Removing the individual with smallest contribution to diversity is used in [25] who consider $(\mu + \lambda)$ EA with λ up to 10. It is worth mentioning that subset selection has gained increasing attention within the evolutionary multi-objective optimisation (EMO) community in recent years [27].

A. Our Contribution

Most studies in the EDO literature work with a $(\mu+1)$ EA, where a single solution is generated in each iteration. A solution with the least contribution to the population diversity is then discarded, and the EA proceeds to the next generation. We investigate EDO algorithms with an offspring population size greater than 1 and explore its potential benefit. Accordingly, we aim to answer the following two questions: (1) How does the use of a $(\mu+\lambda)$ EA affect the diversity of solutions? (2) What is an efficient way to select the next generation in EDO? To answer these questions, we introduce two survival selection methods: (1) a tournament-based subset selection method, where subsets of solutions from the offspring and the parent population are iteratively selected and compete with each other, and the solution with the least contribution is discarded until μ solutions remain; the key point here is that after each deletion, the contribution of the remaining solutions must be updated, and (2) an EA-based subset selection method, where an inner EA is introduced to determine the next generation; we compare these methods with the greedy approach in [25] where the solutions with the least contribution are removed one by one until μ solutions remain. We incorporate a mechanism into the EA-based selection method that takes into account the length of solutions in addition to their contribution to diversity, helping to avoid local diversity optima. We conduct a comprehensive empirical study to compare these methods with the conventional $(\mu+1)$ EA used in the literature. Experiments show that EA-based selection outperforms the other methods in most instances.

The remainder of this paper is structured as follows: Section II defines EDO and the TSP problem as the benchmark case study. Section III presents the $(\mu+\lambda)$ EA and introduces three survival selection approaches for the algorithm. IV compares $(\mu+\lambda)$ EA using the selection methods and the conventional $(\mu+1)$ EA thorough a comprehensive experimental investigation. We conclude with some remarks and future research perspectives.

II. Preliminaries

We consider the TSP as the benchmark problem in this study. The TSP is defined on a directed complete graph $G=(V,E)$ where V is a set of $n=|V|$ nodes and E is the set of pairwise edges between the nodes. A non-negative distance (weight) $w(e)$ is associated with each edge $e\in E$. The objective of the TSP is to find a roundtrip tour that visits all the nodes exactly once and returns to the first city (formally, a permutation $\pi=(\pi(1),\dots,\pi(n))$ of the nodes is sought), with minimum distance travelled:

$$c(\pi)=w(\pi(n),\pi(1))+\sum_{i=1}^{n-1} w(\pi(i),\pi(i+1))\rightarrow \min!$$

In this study we consider the symmetric TSP where $w(u,v)=w(v,u)$ holds for all $u,v\in V$. As mentioned, EDO treats the actual objective function of the problem as constraints and employs a diversity measure as the fitness function:

$$\max D(P) \text{ subject to } c(\pi)\leq(1+\alpha)\cdot \text{OPT } \forall\pi\in P.$$

Here, P is a μ-size population of TSP tours, $D(P)$ is a measure to quantify the diversity of the population, OPT is the objective value of an optimal solution for the TSP problem, α denotes a threshold for acceptable objective values for the TSP solutions.

A. Diversity Measure

EDO requires a measure to quantify the diversity of the TSP solutions. Here, we adopt the *entropy* for two reasons: 1) it yields stronger results in TSP benchmark instances compared to distance-based diversity measures [28]. 2) it has been incorporated in several EDO papers [19], [20], [26]. Let $f(e_{ij})$ be the number of tours in P that contain the directed edge e_{ij}. The entropy of a set of tours P is calculated as

$$H(P)=-\sum_{e_{ij}\in P}\left(\frac{f(e_{ij})}{2n\mu}\right)\cdot\ln\left(\frac{f(e_{ij})}{2n\mu}\right)$$

Note that each solution $\pi\in P$ includes $2n$ edges since we consider the symmetric TSP; given that, there are μ tours in P and thus P has $2n\mu$ edges in total. Obviously, a population with the least diversity is when we have μ copies of a single tour in P, and the entropy value in this case is $H_{\min}=\ln(2n)$. The maximum entropy can then be calculated as follows:

$$\begin{aligned}H_{\max}=&-\left((2n\mu)-(f_0\cdot|E|)\right)\cdot\left(\frac{f_0+1}{2n\mu}\right)\cdot\ln\left(\frac{f_0+1}{2n\mu}\right)\\&-\left((f_0+1)\cdot|E|)-(2n\mu)\right)\cdot\left(\frac{f_0}{2n\mu}\right)\cdot\ln\left(\frac{f_0}{2n\mu}\right)\end{aligned}$$

Interested readers are referred to [19] for the proof and further details on $H_{\max}$.

III. EDO with a $(\mu+\lambda)$ EA

Most studies in the EDO literature use a $(\mu+1)$ EA to maximise the diversity of tours, where the offspring can replace either the parent or one of the individuals in the population P (see, e.g., [22] or [23]) since an increase in offspring pool raises the complexity of the problem due to dependency of diversity contribution of the solutions. This section aims to answer two questions: 1) Is it beneficial to increase the pool of offspring? 2) Given the complexity of the diversity problem, what is the best survival selection approach to determine the next generation? To answer these questions, we propose three subset selection methods to determine the next generation in each iteration of the EDO $(\mu+\lambda)$ EA.

Algorithm 1 Diversity maximising $(\mu + \lambda)$ EA

Input: Population size μ, size of offspring pool $\lambda \geq 1$

1: Initialise the population P with μ TSP tours such that $c(\pi) \leq (1+\alpha) \cdot \mathrm{OPT}$ for all $\pi \in P$.
2: Set $Q = \emptyset$.
3: **while** termination condition not met **do**
4:     **while** $|Q| < \lambda$ **do**
5:         Choose $\pi \in P$ uniformly at random and produce an offspring π' of π by mutation.
6:         If $c(\pi') \leq (1+\alpha) \cdot \mathrm{OPT}$, add π' to Q.
7:     Set $P = \texttt{SELECT}(P \cup Q)$. // Subset selection

Algorithm 2 Greedy subset selection.

Input: Multi-set S with $|S| = \mu + \lambda$

while $|S| > \mu$ **do**
    Remove one individual π from S, where $\pi = \arg\max_{q \in S} H(S \setminus \{q\})$.

Algorithm 1 shows the pseudo-code of the EDO $(\mu + \lambda)$ EA with a parameterisable subset selection procedure (see call to function SELECT in line 7 of Algorithm 1). In this study, we employed 2-opt as the mutation operator. The main challenge is, given a population P with $|P| = \mu$, a set of offspring individuals Q with $|Q| = \lambda \geq 1$, to select a subset S^* of the union set $S = P \cup Q$ such that

$$S^* = \underset{S' \subset S, |S'| = \mu}{\arg\max} \; H(S \setminus S').$$

There are $\binom{\mu+\lambda}{\mu}$ possible subsets. As a consequence, iterating over all these subsets is computationally infeasible as the number of subsets grows exponentially if λ is in the regime of εn. We therefore tackle this subset-selection problem by means of three different heuristics.

A. Greedy selection

The greedy selection strategy (see Algorithm 2) is a straight-forward generalisation of the greedy survival selection that has already been used in [25]. In the ith iteration the algorithm removes an individual from the set S whose deletion leads to smallest loss in diversity. Note that this selection method is computationally costly. It requires, assuming that all λ offspring individuals adhere to the quality constraint, $(\mu + \lambda - i)$ H-evaluations in the ith iteration for $i = 0, \ldots, \lambda - 1$ and thus the overall number of H-evaluations is

$$\sum_{i=0}^{\lambda-1} (\mu + \lambda - i) = \lambda \left(\frac{2\mu + \lambda + 1}{2} \right) = \Theta(\lambda\mu + \lambda^2).$$

Note that the $(\mu+1)$ EA adopted in most of EDO studies in the literature is a special case of the $(\mu + \lambda)$ EA in Algorithm 1 with the greedy selection method.

B. Tournament selection

We adopt a tournament selection strategy iteratively (see Algorithm 3). In λ iterations the algorithm samples a subset R with $|R| = r$, $r \geq 2$ being the tournament size, individuals from the set uniformly at random with replacement. Eventually, the individual is dropped whose deletion yields the maximal H-value. The number of H-evaluations is $\lambda \cdot r$ since the H-value is calculated r times per iteration and it takes λ iterations until the set is reduced to μ individuals; this is $\Theta(\lambda)$ for constant r.

Algorithm 3 Tournament subset selection.

Input: Multi-set S, $|S| = \mu + \lambda$, tournament size $r \geq 2$

while $|S| > \mu$ **do**
    Select a subset $R \subseteq S$ of r individuals uniformly at random with replacement.
    Remove one individual π from S, where $\pi = \arg\max_{q \in R} H(S \setminus \{q\})$.

Algorithm 4 $(\mu + \lambda)$ EA-based subset selection.

Input: Multi-set S with $|S| = \mu + \lambda$

Initialise $x \in \{0,1\}^{|S|}$ where $|x|_1 = \mu$.
for L iterations **do**
    Generate y from x by flipping one-bits independently with probability of w/μ, where w is selected based on power of law distribution. Then flip the same number of bits from zero to one randomly to maintain $|y|_1 = |x|_1 = \mu$.
    if $F(y) \geq F(x)$ **then** // Refer to Eq. (1)
        Replace x with y.

C. EA-based selection

The subset-selection process itself is optimised with a $(1+1)$ EA. To this end we maintain a bitstring $x \in \{0,1\}^{\mu+\lambda}$ and optimise the diversity of the set $S' = \{\pi_i \in S \mid x_i = 1\}$ under the equality constraint $|x|_1 = \mu$ (see Algorithm 4). The algorithm is initialised such that exactly μ individuals are selected (i.e., $|x|_1 = \mu$). In L iterations the algorithm generates an offspring y by flipping zero-bits to ones and the same amount of one-bits to zero in the parent x. The number of flipped bits is $2 \cdot l$ where l is sampled from a Binomial distribution with parameters $\mu + \lambda$ and $\frac{w}{n} \in [0,1]$ where w is selected based on a power-law distribution as it is done in the heavy-tailed mutation [29]. This is done to allow the algorithm to escape local optima by increasing the probability to flip multiple ones and zeros. Similar to all EAs, the EA-based selection method requires a fitness function. The most obvious choice is to use the diversity of y as the fitness function. However, focusing solely on $H(y)$ results in a rapid increase in solution length and increases the likelihood of getting stuck in local diversity optima. Therefore, we introduce a mechanism based on tour length to avoid stagnation in local optima. We first calculate H_0 and l_0 which are the diversity of the population P and the summation of length of individuals in P, respectively. The fitness of offspring in Algorithm 4 can be calculated from:

$$F(x) = \begin{cases} \frac{\Delta H}{\Delta l} & \text{if } \Delta H > 0 \text{ and } \Delta l \geq 0 \\ M & \text{if } \Delta H \geq 0 \text{ and } \Delta l < 0 \\ -M & \text{if } \Delta H < 0 \end{cases} \quad (1)$$

where ΔH and Δl are $H(y)-H_0$ and $l(y)-l_0$, respectively. We use F as the fitness function, a higher F, a fitter solution. The first case is when the diversity of the population increased ($\Delta H > 0$) and the total length of the tours also increased or remained unchanged ($\Delta l \geq 0$) which means that the quality of the tours does not improve; then $F(y)$ is $\frac{\Delta H}{\Delta l}$. If the diversity of solutions increases or remains unchanged ($\Delta H \geq 0$) and the quality of the solutions improves ($\Delta l < 0$), then $F(y)$ is a large number M, and we break the search and choose the set y for the next generation. Finally, if diversity decreases, $F(y)$ is equal to a large negative number $-M$ to ensure that we do not select a set that decreases diversity. Note that L is set to $2\mu\lambda$ in this study based on preliminary investigations. Note that if $\Delta H \geq 0$ and $\Delta l < 0$, we stop the search for the iteration and go to the next generation.

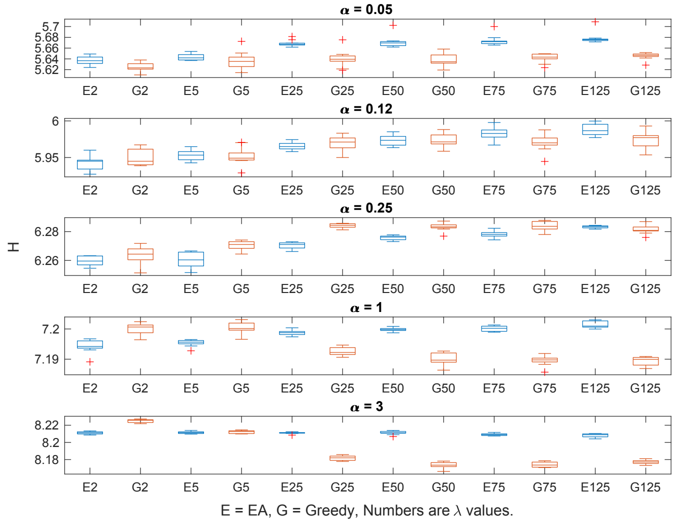

Fig. 1. Distributions of H-values of the final populations based on 10 independent runs for $(\mu+\lambda)$ EA (E) and Greedy (G) subset selection on instance st70. The plots show results for $\alpha \in \{0.02, 0.12, 0.25, 1, 3\}$ (row-wise) and $\lambda \in \{2, 5, 50, 125\}$ (from left to right).

IV. Experimental Investigation

We conduct two series of empirical investigations to study the impact of an offspring pool $\lambda \geq 2$. We first look at the algorithms' performance in the setting of unconstrained diversity optimisation. Then, we analyse constrained diversity optimisation. For the sake of fair comparison, we consider the number of diversity evaluations (H-evaluations) as the termination criterion.

A. *Unconstrained Diversity Optimisation*

We first scrutinise the algorithms' performance in the unconstrained case, where no constraints are imposed on the quality of the tours. We conduct the experiments on a complete graph with $n = 50$ nodes, and consider the following values for the other parameters: $\mu \in \{20, 21, \ldots, 30\}$. The termination criterion and λ are set to 5×10^7 diversity evaluations and 12, respectively, based on the preliminary investigations. According to the pigeonhole principle, the most challenging case in unconstrained diversity optimisation occurs when $2n\mu$ is a factor of $|E|$ (the number of edges). While For $n = 50$, the most difficult cases arise when μ is any factor of 24. Thus, we set $\mu \in \{20, 21, \ldots, 30\}$ to take such challenging cases into account. Table I summarises the results for these experiments. One can observe that the EA-based subset selection algorithm outperforms the competitors followed by the conventional $(\mu + 1)$ EA and the greedy algorithm. The EA algorithm always hits $H_{\max}$ on all instances except where $\mu \in \{24, 25\}$. In these cases, the EA selection hits the optimum, 10 and 16 times, respectively. The $(\mu+1)$ EA cannot bring about the optimal population when $\mu \in \{23, 24, 25\}$ out of 30 runs and it hits the optimum only one time when $\mu = 26$. The $(\mu + 1)$ EA performance gets better as we deviate from $\mu = 24$, and it results in the optimum for the two largest μ values in all 30 independent runs, similar to the EA algorithm. Moreover, the statistical observation confirms a significant difference between the EA algorithm and the $(\mu + 1)$ EA in cases $\mu \in \{23, \ldots, 25\}$. The tournament and the greedy algorithms never resulted in the optimums values in this setting. In fact, the tournament algorithm's results are far away from the optimums.

In conclusion, we can confirm that an offspring pool of $\lambda \geq 2$ can boost the algorithm performance in the unconstrained optimisation if a suitable subset selection method such as the EA-based approach is employed.

B. *Constrained Diversity Optimisation*

We now conduct a series experiments to evaluate the proposed methods' performance in constrained diversity optimisation. In the first set of experiments which the results are summarised in Table II, we test the algorithms on the instances st70, eil101, and a 280 from the TSPlib [30]. We set the tournament size to $r = 3$, the offspring pool size to $\lambda = 50$, and experiments with several values for the quality constraint $\alpha \in \{0.05, 0.12, 0, 25\}$ and the population size $\mu \in \{50, 100\}$. The termination criterion is set to 10^7 H-evaluations. Table II presented in this section reveals some interesting findings. Firstly, the tournament selection method outperforms the other methods when the smallest considered α-value ($\alpha = 0.05$) is used, with the EA algorithm following closely behind. Conversely, the vanilla $(\mu + 1)$ EA performs the worst in these cases. This may be due to its inferior exploration capabilities, as k-tournament selection encourages exploration over exploitation. As α increases to 0.12, the EA-based selection method produces the best results, followed by the greedy and tournament methods. The greedy algorithm

TABLE I

Comparison of the proposed subset selection schemes and $(\mu+1)$ EA in the unconstrained optimisation setting. The column *rate* shows the number of times obtaining the optimal diversity out of 30 independent runs. Stat shows the results of a Kruskal-Wallis test at significance level of 95% with Bonferroni correction. X^+ means the median of the measure is better than the one for variant X, X^- means it is worse and X^* indicates no significant difference.

	$(\mu+\lambda)$ **EA**			**Tournament**			**Greedy**			$(\mu+1)$ **EA**		
μ	H	Stat (1)	rate	H	Stat (2)	rate	H	Stat (3)	rate	H	Stat (4)	rate
20	7.6	$2^{+}3^{+}4^{*}$	**30**	6.83	$1^{-}3^{-}4^{-}$	0	7.49	$1^{-}2^{+}4^{-}$	0	7.6	$1^{*}2^{+}3^{+}$	29
21	7.65	$2^{+}3^{+}4^{*}$	**30**	6.86	$1^{-}3^{-}4^{-}$	0	7.53	$1^{-}2^{+}4^{-}$	0	7.65	$1^{*}2^{+}3^{+}$	18
22	7.7	$2^{+}3^{+}4^{*}$	**30**	6.89	$1^{-}3^{-}4^{-}$	0	7.57	$1^{-}2^{+}4^{-}$	0	7.69	$1^{*}2^{+}3^{+}$	11
23	7.74	$2^{+}3^{+}4^{+}$	**30**	6.92	$1^{-}3^{-}4^{-}$	0	7.6	$1^{-}2^{+}4^{-}$	0	7.73	$1^{-}2^{+}3^{+}$	0
24	7.78	$2^{+}3^{+}4^{+}$	**10**	6.93	$1^{-}3^{-}4^{-}$	0	7.63	$1^{-}2^{+}4^{-}$	0	7.77	$1^{-}2^{+}3^{+}$	0
25	7.8	$2^{+}3^{+}4^{+}$	**16**	6.96	$1^{-}3^{-}4^{-}$	0	7.65	$1^{-}2^{+}4^{-}$	0	7.78	$1^{-}2^{+}3^{+}$	0
26	7.78	$2^{+}3^{+}4^{+}$	**30**	6.98	$1^{-}3^{-}4^{-}$	0	7.67	$1^{-}2^{+}4^{-}$	0	7.78	$1^{-}2^{+}3^{+}$	1
27	7.77	$2^{+}3^{+}4^{*}$	**30**	7	$1^{-}3^{-}4^{-}$	0	7.69	$1^{-}2^{+}4^{-}$	0	7.77	$1^{*}2^{+}3^{+}$	24
28	7.76	$2^{+}3^{+}4^{*}$	**30**	7.03	$1^{-}3^{-}4^{-}$	0	7.7	$1^{-}2^{+}4^{-}$	0	7.76	$1^{*}2^{+}3^{+}$	29
29	7.76	$2^{+}3^{+}4^{*}$	**30**	7.04	$1^{-}3^{-}4^{-}$	0	7.71	$1^{-}2^{+}4^{-}$	0	7.76	$1^{*}2^{+}3^{+}$	**30**
30	7.75	$2^{+}3^{+}4^{*}$	**30**	7.07	$1^{-}3^{-}4^{-}$	0	7.71	$1^{-}2^{+}4^{-}$	0	7.75	$1^{*}2^{+}3^{+}$	**30**

TABLE II

Comparison of the proposed subset selection schemes and $(\mu+1)$ EA. The termination criterion is 10^7 H-evaluations. The notations are in line with Table I

			EA		**Tournament**		**Greedy**		$(\mu+1)$ **EA**	
Instance	α	μ	H	Stat (1)	H	Stat (2)	H	Stat (3)	H	Stat (4)
st70	0.05	50	5.67	$2^{-}3^{+}4^{+}$	**5.7**	$1^{+}3^{+}4^{+}$	5.64	$1^{-}2^{-}4^{+}$	5.62	$1^{-}2^{-}3^{-}$
st70	0.05	100	5.7	$2^{*}3^{+}4^{+}$	**5.73**	$1^{*}3^{+}4^{+}$	5.65	$1^{-}2^{-}4^{*}$	5.65	$1^{-}2^{-}3^{*}$
st70	0.12	50	**5.98**	$2^{+}3^{*}4^{+}$	5.95	$1^{-}3^{-}4^{+}$	5.97	$1^{*}2^{+}4^{+}$	5.93	$1^{-}2^{-}3^{-}$
st70	0.12	100	**6.01**	$2^{+}3^{+}4^{+}$	5.98	$1^{-}3^{*}4^{+}$	5.98	$1^{-}2^{*}4^{+}$	5.96	$1^{-}2^{-}3^{-}$
st70	0.25	50	**6.28**	$2^{+}3^{*}4^{+}$	6.2	$1^{-}3^{-}4^{-}$	**6.28**	$1^{*}2^{+}4^{+}$	6.25	$1^{-}2^{+}3^{-}$
st70	0.25	100	**6.3**	$2^{+}3^{*}4^{+}$	6.25	$1^{-}3^{-}4^{-}$	**6.3**	$1^{*}2^{+}4^{+}$	6.28	$1^{-}2^{+}3^{-}$
eil101	0.05	50	6.02	$2^{-}3^{+}4^{+}$	**6.04**	$1^{+}3^{+}4^{+}$	5.97	$1^{-}2^{-}4^{+}$	5.93	$1^{-}2^{-}3^{-}$
eil101	0.05	100	6.05	$2^{-}3^{+}4^{+}$	**6.06**	$1^{+}3^{+}4^{+}$	5.98	$1^{-}2^{-}4^{*}$	5.97	$1^{-}2^{-}3^{*}$
eil101	0.12	50	**6.35**	$2^{+}3^{+}4^{+}$	6.31	$1^{-}3^{-}4^{*}$	6.34	$1^{-}2^{+}4^{+}$	6.3	$1^{-}2^{*}3^{-}$
eil101	0.12	100	**6.37**	$2^{+}3^{+}4^{+}$	6.34	$1^{-}3^{*}4^{+}$	6.34	$1^{-}2^{*}4^{+}$	6.33	$1^{-}2^{-}3^{-}$
eil101	0.25	50	**6.73**	$2^{+}3^{*}4^{+}$	6.63	$1^{-}3^{-}4^{-}$	**6.73**	$1^{*}2^{+}4^{+}$	6.7	$1^{-}2^{+}3^{-}$
eil101	0.25	100	**6.75**	$2^{+}3^{+}4^{+}$	6.68	$1^{-}3^{-}4^{-}$	6.74	$1^{-}2^{+}4^{+}$	6.73	$1^{-}2^{+}3^{-}$
a280	0.05	50	6.87	$2^{-}3^{+}4^{+}$	**6.88**	$1^{+}3^{+}4^{+}$	6.84	$1^{-}2^{-}4^{+}$	6.82	$1^{-}2^{-}3^{-}$
a280	0.05	100	6.89	$2^{-}3^{+}4^{+}$	**6.9**	$1^{+}3^{+}4^{+}$	6.85	$1^{-}2^{-}4^{*}$	6.84	$1^{-}2^{-}3^{*}$
a280	0.12	50	**7.15**	$2^{+}3^{+}4^{+}$	7.11	$1^{-}3^{-}4^{+}$	7.13	$1^{-}2^{+}4^{+}$	7.1	$1^{-}2^{-}3^{-}$
a280	0.12	100	**7.16**	$2^{+}3^{+}4^{+}$	7.15	$1^{-}3^{*}4^{+}$	7.14	$1^{-}2^{*}4^{+}$	7.14	$1^{-}2^{-}3^{-}$
a280	0.25	50	**7.49**	$2^{+}3^{+}4^{+}$	7.4	$1^{-}3^{-}4^{-}$	7.48	$1^{-}2^{+}4^{+}$	7.45	$1^{-}2^{+}3^{-}$
a280	0.25	100	7.45	$2^{*}3^{-}4^{-}$	7.45	$1^{*}3^{-}4^{-}$	**7.49**	$1^{+}2^{+}4^{*}$	7.48	$1^{+}2^{+}3^{*}$

produces sets with slightly higher diversity when compared to tournament selection. However, the $(\mu+1)$ EA performs poorly in this case. For $\alpha = 0.25$, the greedy algorithm performs best for $\mu = 50$, while the EA outperforms the others for $\mu = 100$, with the $(\mu+1)$ EA ranking closely behind. The tournament selection method performs worst in these cases. Overall, the EA exhibits the most stable performance across all instances, always ranking as the best or second-best method. This may be attributed to the balance between exploration and exploitation that the EA offers. Selection methods with more exploration appear to perform better with tight quality constraints, whereas algorithms that require more exploitation perform better when α is large.

In the next series of experiments, we investigate the performance of the EA and greedy methods on st70 when $\alpha \in \{0.02, 0.12, 0.25, 1, 3\}$ and $\lambda \in \{2, 5, 50, 125\}$. Figure 1 shows that the performance of the EA is slightly better for larger λ-values $(75, 125)$, while it strongly depends on the α-value for the greedy approach. The greedy methods preforms slightly better with large λ when α is small

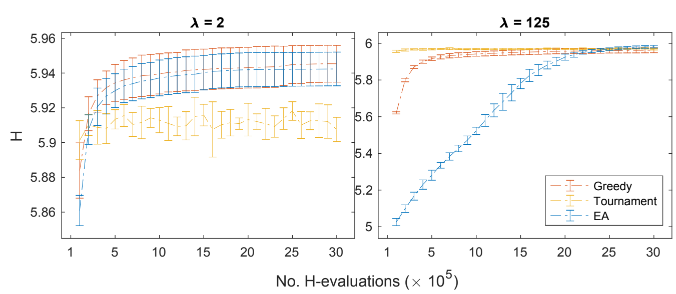

Fig. 2. Representative trajectories of the proposed subset selection methods on instance st70 with $\alpha = 0.12$ and $\lambda \in \{2, 125\}$.

(0.05), it is the other way around in the cases of large α-values ($\alpha = 3$). As $\lambda \to 1$, the greedy algorithm becomes equivalent to the vanilla $(\mu + 1)$ EA. We observed that $(\mu + 1)$ EA outperforms the greedy method in unconstrained diversity. One could guess that smaller values of λ works better for the greedy method for relaxed constraints. In general, the EA's performance is more stable and gets less affected by different values of α, μ, and λ.

Figure 2 illustrates representative trajectories of the proposed selection methods on st70 where $\alpha = 0.12$ and $\lambda = \{2, 125\}$. The tournament selection has the fastest convergence, and increase in λ affects the convergence pace less compared to the other methods. The increase in λ has the most significant impact on the EA's convergence pace since there is linear relation between L and λ (we set $L = 2\mu\lambda$). Therefore, when λ increases, more H-evaluations are spent on each loop of the EA selection. The increase in λ also makes an impact on the greedy convergence pace, but not as severely as it does on the EA selection.

Conclusion

We studied survival selection methods for EDO working with an offspring population size greater than 1. We introduced three subset selection methods that enable us replacing more than one individuals with offspring for the next generation. Our results show that an EA-based method as a sub-procedure of a $(\mu + \lambda)$ EA with $\lambda \geq 2$ outperforms the standard $(\mu + 1)$ EA and the other proposed methods in most cases.

Acknowledgements

This work was supported by the Australian Research Council through grant FT200100536.

References

[1] X. Li, M. G. Epitropakis, K. Deb, and A. P. Engelbrecht, "Seeking multiple solutions: An updated survey on niching methods and their applications," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 518–538, 2017.

[2] A. Cully and J. Mouret, "Behavioral repertoire learning in robotics," in *GECCO*. ACM, 2013, pp. 175–182.

[3] J. Clune, J. Mouret, and H. Lipson, "Summary of "the evolutionary origins of modularity"," in *GECCO (Companion)*. ACM, 2013, pp. 23–24.

[4] J. K. Pugh, L. B. Soros, P. A. Szerlip, and K. O. Stanley, "Confronting the challenge of quality diversity," in *GECCO*. ACM, 2015, pp. 967–974.

[5] J. K. Pugh, L. B. Soros, and K. O. Stanley, "Quality diversity: A new frontier for evolutionary computation," *Frontiers Robotics AI*, vol. 3, p. 40, 2016.

[6] A. Nikfarjam, A. V. Do, and F. Neumann, "Analysis of quality diversity algorithms for the knapsack problem," in *PPSN (2)*, ser. Lecture Notes in Computer Science, vol. 13399. Springer, 2022, pp. 413–427.

[7] J. Bossek and F. Neumann, "Exploring the feature space of TSP instances using quality diversity," in *GECCO*. ACM, 2022, pp. 186–194.

[8] A. Nikfarjam, T. Stanford, A. Neumann, D. Dumuid, and F. Neumann, "Quality diversity approaches for time-use optimisation to improve health outcomes," in *GECCO*. ACM, 2024.

[9] L. Baraton, L. Urbano, Annafedericaand Brevault, and M. Balesdent, "Bayesian quality-diversity optimization forconditional search-space problems," *Optimization and Engineering*, pp. 1–46, 2025.

[10] B. Tjanaka, M. C. Fontaine, J. Togelius, and S. Nikolaidis, "Approximating gradients for differentiable quality diversity in reinforcement learning," in *GECCO*. ACM, 2022, pp. 1102–1111.

[11] S. Earle, J. Snider, M. C. Fontaine, S. Nikolaidis, and J. Togelius, "Illuminating diverse neural cellular automata for level generation," in *GECCO*. ACM, 2022, pp. 68–76.

[12] T. Ulrich and L. Thiele, "Maximizing population diversity in single-objective optimization," in *GECCO*. ACM, 2011, pp. 641–648.

[13] B. Alexander, J. Kortman, and A. Neumann, "Evolution of artistic image variants through feature based diversity optimisation," in *GECCO*. ACM, 2017, pp. 171–178.

[14] W. Gao, S. Nallaperuma, and F. Neumann, "Feature-based diversity optimization for problem instance classification," *Evol. Comput.*, vol. 29, no. 1, pp. 107–128, 2021.

[15] A. Neumann, W. Gao, C. Doerr, F. Neumann, and M. Wagner, "Discrepancy-based evolutionary diversity optimization," in *GECCO*. ACM, 2018, pp. 991–998.

[16] A. Neumann, W. Gao, M. Wagner, and F. Neumann, "Evolutionary diversity optimization using multi-objective indicators," in *GECCO*. ACM, 2019, pp. 837–845.

[17] J. Bossek, P. Kerschke, A. Neumann, M. Wagner, F. Neumann, and H. Trautmann, "Evolving diverse TSP instances by means of novel and creative mutation operators," in *FOGA*. ACM, 2019, pp. 58–71.

[18] A. V. Do, J. Bossek, A. Neumann, and F. Neumann, "Evolving diverse sets of tours for the travelling salesperson problem," in *GECCO*. ACM, 2020, pp. 681–689.

[19] A. Nikfarjam, J. Bossek, A. Neumann, and F. Neumann, "Entropy-based evolutionary diversity optimisation for the traveling salesperson problem," in *GECCO*. ACM, 2021, pp. 600–608.

[20] ——, "Computing diverse sets of high quality TSP tours by eax-based evolutionary diversity optimisation," in *FOGA*. ACM, 2021, pp. 9:1–9:11.

[21] A. V. Do, M. Guo, A. Neumann, and F. Neumann, "Analysis of evolutionary diversity optimization for permutation problems," *ACM Trans. Evol. Learn. Optim.*, vol. 2, no. 3, pp. 11:1–11:27, 2022.

[22] J. Bossek and F. Neumann, "Evolutionary diversity optimization and the minimum spanning tree problem," in *GECCO*. ACM, 2021, pp. 198–206.

[23] A. Neumann, J. Bossek, and F. Neumann, "Diversifying greedy sampling and evolutionary diversity optimisation for constrained monotone submodular functions," in *GECCO*. ACM, 2021, pp. 261–269.

[24] A. Nikfarjam, A. Neumann, and F. Neumann, "Evolutionary diversity optimisation for the traveling thief problem," in *GECCO*. ACM, 2022, pp. 749–756.

[25] A. Neumann, S. Gounder, X. Yan, G. Sherman, B. Campbell, M. Guo, and F. Neumann, "Evolutionary diversity optimization for the detection and concealment of spatially defined communication networks," in *GECCO*. ACM, 2023.

[26] A. Nikfarjam, R. Rothenberger, F. Neumann, and T. Friedrich, "Evolutionary diversity optimisation in constructing satisfying assignments," in *GECCO*. ACM, 2023.

[27] Y. Gu, C. Bian, M. Li, and C. Qian, "Subset selection for evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 28, no. 2, pp. 403–417, 2024.

[28] Y. Nagata, "High-order entropy-based population diversity measures in the traveling salesman problem," *Evol. Comput.*, vol. 28, no. 4, pp. 595–619, 2020.

[29] B. Doerr, H. P. Le, R. Makhmara, and T. D. Nguyen, "Fast genetic algorithms," in *GECCO*. ACM, 2017, pp. 777–784.

[30] G. Reinelt, "TSPLIB–a traveling salesman problem library," *ORSA Journal on Computing*, vol. 3, no. 4, pp. 376–384, 1991.