Evolving Diverse Sets of Tours for the Travelling Salesperson Problem

Anh Viet Do
Optimisation and Logistics
The University of Adelaide, Adelaide, Australia

Jakob Bossek
Optimisation and Logistics
The University of Adelaide, Adelaide, Australia

Aneta Neumann
Optimisation and Logistics
The University of Adelaide, Adelaide, Australia

Frank Neumann
Optimisation and Logistics
The University of Adelaide, Adelaide, Australia

ABSTRACT

Evolving diverse sets of high quality solutions has gained increasing interest in the evolutionary computation literature in recent years. With this paper, we contribute to this area of research by examining evolutionary diversity optimisation approaches for the classical Traveling Salesperson Problem (TSP). We study the impact of using different diversity measures for a given set of tours and the ability of evolutionary algorithms to obtain a diverse set of high quality solutions when adopting these measures. Our studies show that a large variety of diverse high quality tours can be achieved by using our approaches. Furthermore, we compare our approaches in terms of theoretical properties and the final set of tours obtained by the evolutionary diversity optimisation algorithm.

CCS CONCEPTS

• **Theory of computation** → **Evolutionary algorithms**; • **Mathematics of computing** → *Graph theory*.

KEYWORDS

Evolutionary algorithms, diversity maximisation, travelling salesperson problem

ACM Reference Format:
Anh Viet Do, Jakob Bossek, Aneta Neumann, and Frank Neumann. 2020. Evolving Diverse Sets of Tours for the Travelling Salesperson Problem. In *Genetic and Evolutionary Computation Conference (GECCO '20), July 8–12, 2020, Cancún, Mexico.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3377930.3389844

1 INTRODUCTION

Evolutionary diversity optimisation aims to construct a set of diverse solutions that all have high quality but differ with respect to important properties of the solutions. This area of research started by Ulrich and Thiele [17, 18] has recently gained significant attention within the evolutionary computation community. It equips practitioners with high quality solutions of variable designs. Furthermore, the methods developed can be used in the context of machine learning and algorithm selection and configuration. Here, evolutionary diversity optimisation is used to evolve a diverse set of instances that can be used for training prediction models that forecast algorithm performance on a newly given instance.

Recent studies in this area of research mainly focused on the diversity measure to be used in the selection process when characterising solutions by underlying features. Differences in feature values as well as weighted combinations of two or more features have been examined in the context of Travelling Salesperson Problem (TSP) instances and images [1, 5]. More recently the use of the discrepancy measure as well as the use of popular indicators from the area of evolutionary multi-objective optimisation have been proposed and evaluated [13, 14]. Special mutation operators for creating diverse sets of TSP instances have been investigated in [3].

In this paper, we examine, for the first time, TSP in the context of evolutionary diversity optimisation. Our focus is on the diversity of the tours themselves, not their qualities. The TSP has been subject to a wide range of studies. Various evolutionary algorithms and other heuristic search methods as well as exact solvers have been developed over the years [6, 10, 12, 20]. Furthermore, the TSP has been studied widely in the area of algorithm selection and configuration (see [8] for a recent overview on this research area) and a wide range of studies regarding important features of TSP instances and their relation to algorithm performance have been carried out [7, 9, 11, 16]. Evolutionary diversity optimisation has so far only been considered for evolving TSP instances [5, 13, 14], but not for computing a diverse set of high quality tours for a given TSP instance.

In this paper, we examine ways to evolve a diverse set of high quality TSP tours for a given TSP instance and population size. A crucial aspect in our study is how to measure the diversity of a tours population for an effective diversity-driven approach. To establish diversity-oriented evolutionary pressure, we propose two diversity measures: an edge distribution diversity measure and a pairwise dissimilarity measure. We study both measures with respect to their properties in the context of evolutionary diversity optimisation and carry out experimental investigations showing how evolutionary algorithms generate diverse populations using the two measures on unweighted TSP instances where all tours are accepted. Our results for this basic set up shows that these cases can be handled effectively and set the basis for studies on classical TSPlib instances [15] where tours are filtered based on qualities.

GECCO '20, July 8–12, 2020, Cancún, Mexico
© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-7128-5/20/07…$15.00
https://doi.org/10.1145/3377930.3389844

We investigate the introduced evolutionary diversity optimisation approach using the two diversity measures on classical instances, and explore the differences in results when using classical k-opt operations where $k = 2, 3, 4$ for a wide range of quality thresholds imposed on the tours. Furthermore, we study how the population size which determines the size of the set of tours effects the ability of the evolutionary diversity optimisation approaches to obtain a high diversity score.

The paper is structured as follows. In Section 2, we introduce the Traveling Salesperson Problem in the context of evolutionary diversity optimisation and the algorithms that are part of our investigations. In Section 3, we introduce the edge diversity measure and investigate its theoretical properties. Section 4 introduces the pairwise distance diversity measure along with its theoretical properties. We report on our experimental investigations in Section 5 and finish with some conclusions.

2 MAXIMISING DIVERSITY IN TSP

The Traveling Salesperson problem (TSP), one of the best-known $\mathcal{NP}$-hard combinatorial optimisation problems, can be described as follows: Given a complete graph $G = (V, E)$ with $n = |V|$ cities, $m = n(n-1)/2 = |E|$ edges and the pairwise distances between the cities, the goal is to compute a tour of minimal length that visits each city exactly once and finally returns to the original city. For the Euclidean TSP all cities lie in the Euclidean plane and the pairwise distances between the cities are determined by the Euclidean metric. Let $V = \{1, \ldots, n\}$. The goal is to find a permutation $\pi : V \to V$ that minimises the cost function

$$c(\pi) = d(\pi(n), \pi(1)) + \sum_{i=1}^{n-1} d(\pi(i), \pi(i+1)),$$

where $d(i, j)$ is the Euclidean distance between points i and j. Note that the Euclidean TSP remains an $\mathcal{NP}$-hard combinatorial optimisation problem.

In this paper, we consider diversity optimisation for the Traveling Salesperson Problem. For each TSP instance, our goal is to find a set P of $\mu = |P|$ tours that is diverse with respect to some diversity measure, while each tour meets a given quality threshold. Let I be an individual (which constitutes a permutation of the given n cities) and $c(I)$ be the cost of I. The quality threshold is met iff $c(I) \leq (1 + \alpha) \cdot OPT$, where OPT is the value of an optimal tour and $\alpha > 0$ is a parameter that determines the required quality of a desired solution. The quality criterion means that the quality threshold is met iff I is a $(1+\alpha)$ approximation of an optimal solution. We assume that the optimal tour is known for a given TSP instance.

In order to optimise the diversity for the Traveling Salesperson Problem we employ a $(\mu + 1)$-EA that has already been used in the context of evolutionary diversity optimisation [1, 3, 5, 13, 14]. Our approach differs from the previous ones in terms of the considered problem and underlying diversity measure that drives the optimisation approach.

We use Algorithm 1 to compute a diverse population consisting of TSP tours where each individual/tour I has to meet a given quality criteria $c(I)$ according to a given threshold. Initially, the population P is generated with μ individuals, and exactly one offspring I' is produced in each iteration. If the offspring I' does not satisfy $c(I') \leq (1+\alpha) \cdot OPT$, then it is discarded. Otherwise I' is added to the population. Afterwards, elitist survival selection is performed with respect to a diversity measure D. For our investigations, we consider some function indicating overlaps between tours $D\colon P \to \mathbb{R}$, which should be minimized to improve diversity. Thus, an individual $I \in P$ is removed such that $D(P \setminus \{I\})$ is minimal among all individuals $J \in P$.

Algorithm 1: Diversity maximising $(\mu + 1)$-EA

```
1 Initialise the population $P$ with $\mu$ TSP tours such that
    $c(I) \leq (1 + \alpha) \cdot OPT$ for all $I \in P$.
2 Choose $I \in P$ uniformly at random and produce an offspring
    $I'$ of $I$ by mutation.
3 If $c(I') \leq (1 + \alpha) \cdot OPT$, add $I'$ to $P$.
4 If $|P| = \mu + 1$, remove exactly one individual $I$, where
    $I = \arg\min_{J \in P} D(P \setminus \{J\})$, from $P$.
5 Repeat steps 2 to 4 until a termination criterion is reached.
```

We introduce two diversity measures for evolutionary diversity optimisation for TSP, namely an *edge diversity* (ED) optimisation approach in Section 3 and a *pairwise edge distances* (PD) optimisation approach in Section 4. We consider vector functions instead of traditional scalar functions, with the hope to capture more nuances in the survival selection mechanism elegantly.

The edge diversity optimisation approach attempts to maximise population diversity without relying on dissimilarities between tours. Instead, it considers how frequent each edge is present in the population, and aims to equalise these frequencies. The goal is a population of tours containing every edge, each as close to k times as possible for some k. The approach makes sense if edges can be considered independent of each other, meaning each edge is present in the population at a frequency independent of that of other edges. This is not true for tours, thus important information may potentially be left out.

On the other hand, the pairwise edge distances approach considers solely the edge distances between all pairs of tours in the population in terms of overlap. It attempts to simultaneously maximise these distances and equalising them, with an emphasis on pairs with the least distances. In effect, it tries to increase these small distances by moving tours away from their closest tours and possibly closer to further tours, using mutation operators. Consequently, it tends to generally increase the dissimilarities between a tour I and the rest $P \setminus \{I\}$. This helps lessen the clustering phenomenon, where tours in the population form low-diversity clusters.

The evolutionary diversity algorithm based on edge diversity measure is compared to the evolutionary diversity optimisation approach based on pairwise edge distances measure in two different settings, using simple unconstrained TSP tours and TSPlib instances [15] in Section 5.

3 MAXIMISING EDGE DIVERSITY

In this approach, we consider diversity in terms of equal representations of edges by tours in the population, or *edge diversity*. It takes into account, for each edge, the number of tours containing it, among the μ solutions in the population. These numbers are referred to as *edge counts*. Given a population P and an edge $e \in E$, we denote by $n(e, P)$ its edge count, which is defined,

$$n(e, P) := |\{T \in P \mid e \in E(T)\}| \in \{0, \dots, \mu\}$$

where $E(T) \subset E$ is the set of edges used by solution T. Then in order to maximise the edge diversity we aim to minimise the vector

$$\mathcal{N}(P) = \mathrm{sort}\,(n(e_1, P), n(e_2, P), \dots, n(e_m, P))$$

in the lexicographic order where sorting is performed in descending order. This is based on the idea of maximising genotypic diversity as the mean of pairwise edge distances [21]. Since μ is fixed, we change the mean to a sum to simplify the function

$$gtype(P) = \sum_{T_1 \in P} \sum_{T_2 \in P} |E(T_1) \setminus E(T_2)|.$$

The maximum edge distance between two different tours is n, so the maximum diversity is $\mu(\mu-1)n$. Let $n_i = n(e_i, P)$. There are n_i tours in P sharing edge e_i. Therefore, it affects $n_i(n_i - 1)/2$ pair-wise edge distances, reducing each by 1. Since they can be added up independently across all edges, the diversity measure is then

$$gtype(P) = \mu(\mu - 1)n + \sum_i n_i - \sum_i n_i^2.$$

Since $\sum_i n_i = \mu n$ is constant, maximising diversity is reduced to minimising $\sum_i n_i^2$. Given $\sum_i n_i$ being constant, the Cauchy–Schwarz inequality implies that $\sum_i n_i^2$ is the smallest when all n_i are as close to being equal to each other as possible. The population with such property minimises $\mathcal{N}$, meaning

$$\underset{P}{\mathrm{argmax}}\{gtype(P)\} = \underset{P}{\mathrm{argmin}}\{\mathcal{N}(P)\}.$$

For complete graphs, the optima for $\mathcal{N}(P)$ can be determined based on the following result, so the maximum diversity can be calculated.

Theorem 1. *For every pair of integers $\mu \geq 1$ and $n \geq 3$, given a complete graph $G = (V, E)$ where $|V| = n$, there is a μ-size population P of tours such that*

$$\max_{e \in E}\{n(e, P)\} - \min_{e \in E}\{n(e, P)\} \leq 1.$$

Proof. We prove this by defining a way to construct such a population. According to Theorem 1 in [2], in every complete graph with $n \geq 3$ vertices, there is a set of $h = \left\lfloor \frac{n-1}{2} \right\rfloor$ pairwise edge-disjoint Hamiltonian cycles. We denote by H the set of all such Hamiltonian cycles in G, and $E(H)$ the set of all edges H contains. We consider 2 cases: n is odd, and n is even.

Assuming n is odd, then G can be decomposed completely into edge-disjoint tours, meaning $E(H) = E$. Let $\mu = kh + r$ where $r \in [0, h)$ and $r \equiv \mu \bmod h$. We construct a population P' by adding all tours in H, each exactly k times. We then construct P from P' by adding all tours in L for any $L \subset H$ where $|L| = r$. With this, $n(e, P) = k$ for all $e \notin E(L)$, and $n(e, P) = k + 1$ for all $e \in E(L)$.

Assuming n is even, then according to [2], for any perfect matching M in G, the sub-graph $G^* = (V, E \setminus M)$ can be decomposed completely into edge-disjoint tours. This means $E(H) = E \setminus M$ for some perfect matching M in G. Let T be the tour in G that goes through all edges in M, $M' = E(T) \setminus M$ be another perfect matching edge-disjoint with M, H' be the set of all edges-disjoint tours in $G' = (V, E \setminus M')$, and $\mu = k(h + 1) + r$ where $r \in [0, h]$ and $r \equiv \mu \bmod (h + 1)$. We construct P with the following steps:

(1) Add all tours in H, each $\left\lfloor \frac{k+1}{2} \right\rfloor$ times
(2) Add all tours in H', each $\left\lfloor \frac{k}{2} \right\rfloor$ times
(3) Add tour T k times
(4) Add all tours in L where $L \subseteq H$ if k is even, or $L \subseteq H'$ otherwise, such that $|L| = r$.

The result is a population P such that $|P| = \mu$. If k is odd, $n(e, P) = k + 1$ for all $e \in E(L) \cup M'$ and $n(e, P) = k$ otherwise. If k is even and positive, $n(e, P) = k + 1$ for all $e \in E(L) \cup M$ and $n(e, P) = k$ otherwise. If $k = 0$, $n(e, P) = 1$ for all $e \in E(L)$ and $n(e, P) = 0$ otherwise. □

As demonstrated by the proof, a drawback of this approach is that it does not necessarily prevent duplication of tours in P when μ is sufficiently large. This is because the sum of $L1$-norm distances is no more sensitive to small distances than large distances. Consequently, the diversity score based on it can still be large when the population consists of low-diversity sub-populations that are highly dissimilar. In other words, this approach is susceptible to clustering.

In the context of EAs, this approach formulates a survival selection mechanism: removing from the population the individual $I \in \mathrm{argmin}_{I \in P}\{\mathcal{N}(P \setminus \{I\})\}$. For an efficient implementation, we consider an equivalent fitness function for each individual $I \in P$

$$n_P(I) = \mathrm{sort}\,((n(e, P))_{e \in I})$$

with descending sorting order. The survival selection mechanism then removes $I \in \mathrm{argmax}_{I \in P}\{n_P(I)\}$. The equivalence can be shown. Since elements of $\mathcal{N}(P \setminus \{I\})$ and $n_P(I)$ are in descending order, they can each be uniquely defined by a vector $(m_i^I)_{i=1,\dots,\mu}$ and $(n_j^I)_{j=1,\dots,\mu}$, respectively, where m_i^I and n_i^I are numbers of elements in $\mathcal{N}(P \setminus \{I\})$ and $n_P(I)$ equal to i, respectively. For $X, Y \in P$, if $\mathcal{N}(P \setminus \{X\}) < \mathcal{N}(P \setminus \{Y\})$ lexicographically, then there must be $j \in [1, \mu]$ such that $m_j^X < m_j^Y$ and $m_i^X = m_i^Y$ for all $i \in (j, \mu]$. Fewer elements equal to j in $\mathcal{N}(P \setminus \{X\})$ must be the consequence of removing more of them, meaning $n_j^X > n_j^Y$ and $n_i^X = n_i^Y$ for all $i \in (j, \mu]$. This implies that $n_P(X) > n_P(Y)$ lexicographically. On the other hand, it is trivial to prove that $\mathcal{N}(P \setminus \{X\}) = \mathcal{N}(P \setminus \{Y\})$ implies $n_P(X) = n_P(Y)$. This means

$$\underset{I \in P}{\mathrm{argmin}}\{\mathcal{N}(P \setminus \{I\})\} = \underset{I \in P}{\mathrm{argmax}}\{n_P(I)\}.$$

With this method, the survival selection mechanism is consisted of three phases:

(1) Calculating the edge counts table: $O(\mu n)$
(2) Calculating $n_P(I)$ for all I: $O(\mu n \log n)$
(3) Finding and removing $I = \mathrm{argmax}_{I \in P}\{n_P(I)\}$: $O(\mu n)$

The complexity of the survival selection is then $O(\mu n \log n)$. On the other hand, the mechanism's complexity when using $gtype(P \setminus \{I\})$ as fitness values is $O(\mu^2 n + \mu^3)$ ($O(\mu^2 n)$ from calculating the edge distances table and $O(\mu^2)$ from calculating $gtype$ from said table for each tour). Asymptotically speaking, our proposal can be faster per iteration in many cases.

4 EQUALISING PAIRWISE EDGE DISTANCES

One way to remedy the clustering phenomenon is to increase edge distances between highly similar tours, potentially at the cost of decreasing edge distances between highly dissimilar tours. This method essentially equalises the tours' pairwise edge distances. As such, we present another approach that discourages clustering by emphasising uniform pairwise edge distances while maximising diversity. We reformulate the edge distance as edge overlap

$$o_{XY} = |E(X) \cap E(Y)| = n - |E(X) \setminus E(Y)|, \forall X, Y \in P.$$

The aim is then to minimise the vector

$$\mathcal{D}(P) = \text{sort}\left((o_{XY})_{X,Y \in P}\right)$$

in the lexicographic order where sorting is performed in descending order. This approach simultaneously maximises diversity via minimising $\sum_{X,Y \in P} o_{XY}$, while also maximising uniformity via equalising o_{XY}. It can be the case that $\mathcal{D}(P) < \mathcal{D}(P')$ and $gtype(P) < gtype(P')$.

Example 1. *Consider a problem instance with a complete graph of* $n = 5$ *vertices and* $\mu = 3$. *Let there be four tours* $T_1 = (1, 3, 5, 4, 2)$, $T_2 = (1, 5, 4, 3, 2)$, $T_3 = (1, 2, 5, 3, 4)$ *and* $T_4 = (1, 5, 2, 3, 4)$, *and two populations* $P_1 = \{T_1, T_2, T_3\}$ *and* $P_2 = \{T_1, T_2, T_4\}$, *we have*

- $gtype(P_1) = 18 < gtype(P_2) = 20$
- $\mathcal{D}(P_1) = (2, 2, 2) < \mathcal{D}(P_2) = (3, 2, 0)$.

It is clear that P_1 *maximises edge distances uniformity and* P_2 *maximises gtype. Furthermore, we can see that among all* 3*-size populations of maximum gtype,* P_2 *maximises edge distances uniformity. Likewise, among all* 3*-size populations of maximum edge distances uniformity,* P_1 *maximises gtype. This example shows that maximising gtype and edge distances uniformity at the same time can be non-trivial.*

In such cases, it is likely that o_{XY} are more equalised in P than in P'. Moreover, the more $gtype(P)$ is larger than $gtype(P')$, the more likely that $\mathcal{D}(P) < \mathcal{D}(P')$. This means evolving with lowering $\mathcal{D}(P)$ as the evolutionary force tends to increase the *gtype* score. However, the maximisation performance is potentially compromised by the emphasis on edge distances uniformity, so the *gtype* score can decrease and its maximum is sometimes unreachable. Given the drawback of this diversity measure, this may not be undesirable.

For a problem instance with graph G and integer μ, assuming there is a population P^* with maximum *gtype*

$$2 \sum_{X,Y \in P^*} o_{XY} = \mu(\mu - 1)n - \max_P\{gtype(P)\},$$

and maximum edge distances uniformity

$$\max_{X,Y \in P}\{o_{XY}\} - \min_{X,Y \in P}\{o_{XY}\} \leq 1,$$

then $P^* \in \text{argmin}_P\{\mathcal{D}(P)\}$. When G is complete and $\mu \leq \left\lfloor \frac{n-1}{2} \right\rfloor$, this assumption is true according to the proof for Theorem 1, and o_{XY} are all zero in P^*.

Like the ED approach, this approach determines the survival mechanism of the EA: removing from the population the individual $I \in \text{argmin}_{I \in P}\{\mathcal{D}(P \setminus \{I\})\}$. The same method can be used to derive an efficient implementation of $O(\mu^2 n + \mu^2 \log \mu)$ time complexity for this mechanism. Consider the fitness function

$$d_P(I) = \text{sort}\left((o_{IX})_{X \in P \setminus \{I\}}\right)$$

with descending sorting order, it can be shown that $\text{argmin}_I\{\mathcal{D}(P \setminus \{I\})\} = \text{argmax}_I\{d_P(I)\}$. As before, we can uniquely define $\mathcal{D}(P)$ and $d_P(I)$ with $(m_i)_{i=1,\ldots,n}$ and $(n_i^I)_{i=1,\ldots,n}$, respectively. This implies that $\mathcal{D}(P \setminus \{I\})$ is defined by $(m_i - n_i^I)_{i=1,\ldots,n}$. For $X, Y \in P$, if $\mathcal{D}(P \setminus \{X\}) < \mathcal{D}(P \setminus \{Y\})$ lexicographically, then it must be that there is $j \in [1, n]$ such that $n_j^X > n_j^Y$ and $n_i^X = n_i^Y$ for all $i \in (j, n]$. This means $d_P(X) > d_P(Y)$. Furthermore, $\mathcal{D}(P \setminus \{X\}) < \mathcal{D}(P \setminus \{Y\})$ obviously implies $d_P(X) = d_P(Y)$. Intuitively, this implementation removes I that has the smallest edge distance to $P \setminus \{I\}$, defined by the minimum edge distance between I and all tours in $P \setminus \{I\}$. The lexicographic ordering of vectors allows an elegant way to resolve draw cases: comparing the second minimum distances, the third, the fourth and so on. The result is the guaranteed non-increasing edge distance between I' and $P \setminus \{I'\}$ for all other tours I' as well. The consequence is convergence to a solution population, in which each tour is reasonably dissimilar to the rest. This approach aligns closely with the diversity formulated in [19], with edge distance being the dissimilarity metric. Since μ is fixed, we introduce an additional normalization factor.

$$div(P) = \sum_{T \in P} dist(T, P \setminus \{T\}) = \frac{1}{\mu n} \sum_{T \in P} \min_{X \in P \setminus \{T\}} \{|E(T) \setminus E(X)|\}.$$

Note that the difference between *gtype* and *div* is the focus on the minimum edge distances, the two measures would be the same if *min* were replaced by *sum* or *mean*. We hypothesise that edge distances uniformity is strongly positively correlated to *div*. This is explored further in Section 5.

5 EXPERIMENTAL INVESTIGATIONS

We perform different series of experiments to gain insights into the process of evolving diverse TSP tours. In all experiments our input is a complete graph $G = (V, E)$ with real-valued cost function $w : E \to \mathbb{R}^+$. We consider 6 variants of algorithm 1, differing in their mutation operators and survival selection mechanisms. The mutation operators are 2-OPT, 3-OPT, 4-OPT, and the survival selection mechanisms are based on the two proposed approaches.

5.1 Unconstrained Diversity Optimisation

In this setting our focus is on diverse TSP tours without constraints on tours' qualities. As such, we study our approaches with random initial populations. We experiment with all combinations of $n = \{50, 100, 200, 500\}$ and $\mu = \{3, 10, 20, 50\}$. For each (n, μ) combination, 30 populations are randomly generated as initial populations for all EA variants, controlling for the initialisation factor. Furthermore, with the established guarantee for optima, a termination criterion is such optima are reached. As for minimising $\mathcal{D}(P)$, another criterion

$$\max_{X,Y \in P}\{o_{XY}\} - \min_{X,Y \in P}\{o_{XY}\} \leq 1$$

is added forming a bound on the optima. An additional limit of μn^2 iterations is imposed on the experiments. Also, since optimal *gtype* scores can be calculated, we record the scores in percentages for ease of comparison across all settings.

According to Tables 1, 2, all variants seem to reliably achieve optima in all cases where $\mu \leq \left\lfloor \frac{n-1}{2} \right\rfloor$, except for 4-OPT variants seemingly stuck in local optima when $n = 50$ and $\mu = 20$. In the other hard cases, none ever reached the optima within the time

Table 1: Comparison in terms of diversity (gtype), the number of iterations (#iter) and results of statistical testing (stat) between variants with ED approach in unconstrained cases. The Kruskal-Wallis test and the Bonferroni correction method [4] are used on #iter. X^+ means the measure is larger than the one for variant X, X^- means smaller and X^* means no difference.

n	μ	2-OPT(1)					3-OPT(2)					4-OPT(3)				
		gtype	**std**	**# iter**	**std**	**stat**	**gtype**	**std**	**# iter**	**std**	**stat**	**gtype**	**std**	**# iter**	**std**	**stat**
50	3	100.00%	0.00	104.50	59.19	$2^+, 3^+$	100.00%	0.00	68.60	35.67	$1^-, 3^*$	100.00%	0.00	**54.37**	24.28	$1^-, 2^*$
	10	100.00%	0.00	**1635.80**	485.88	$2^*, 3^*$	100.00%	0.00	1808.43	484.96	$1^*, 3^*$	100.00%	0.00	1736.60	545.36	$1^*, 2^*$
	20	100.00%	0.00	**25383.03**	8594.81	$2^-, 3^-$	99.97%	0.02	49881.60	471.00	$1^+, 3^*$	99.93%	0.02	50000.00	0.00	$1^+, 2^*$
	50	99.99%	0.00	**125000.00**	0.00	$2^*, 3^*$	99.95%	0.00	**125000.00**	0.00	$1^*, 3^*$	99.93%	0.01	**125000.00**	0.00	$1^*, 2^*$
100	3	100.00%	0.00	170.83	98.50	$2^*, 3^+$	100.00%	0.00	137.53	69.76	$1^*, 3^*$	100.00%	0.00	**102.13**	58.69	$1^-, 2^*$
	10	100.00%	0.00	1901.73	632.60	$2^*, 3^+$	100.00%	0.00	1598.80	386.85	$1^*, 3^+$	100.00%	0.00	**1305.63**	343.38	$1^-, 2^-$
	20	100.00%	0.00	**8422.40**	1742.96	$2^-, 3^*$	100.00%	0.00	9902.63	1984.39	$1^+, 3^*$	100.00%	0.00	9414.27	2018.82	$1^*, 2^*$
	50	99.95%	0.00	**500000.00**	0.00	$2^*, 3^*$	99.86%	0.00	**500000.00**	0.00	$1^*, 3^*$	99.82%	0.01	**500000.00**	0.00	$1^*, 2^*$
200	3	100.00%	0.00	401.03	213.80	$2^+, 3^+$	100.00%	0.00	254.80	111.06	$1^-, 3^*$	100.00%	0.00	**197.60**	78.85	$1^-, 2^*$
	10	100.00%	0.00	3350.47	958.53	$2^+, 3^+$	100.00%	0.00	2261.10	570.16	$1^-, 3^*$	100.00%	0.00	**1888.87**	510.91	$1^-, 2^*$
	20	100.00%	0.00	10189.73	2233.72	$2^+, 3^+$	100.00%	0.00	8959.07	2242.66	$1^-, 3^*$	100.00%	0.00	**7901.73**	1286.07	$1^-, 2^*$
	50	100.00%	0.00	**76856.37**	14056.12	$2^-, 3^-$	100.00%	0.00	102039.53	17118.06	$1^+, 3^-$	100.00%	0.00	132837.73	25315.99	$1^+, 2^+$
500	3	100.00%	0.00	974.33	509.02	$2^+, 3^+$	100.00%	0.00	631.20	267.24	$1^-, 3^*$	100.00%	0.00	**560.77**	332.16	$1^-, 2^*$
	10	100.00%	0.00	6608.33	2298.31	$2^*, 3^+$	100.00%	0.00	5281.23	1729.61	$1^*, 3^+$	100.00%	0.00	**3537.07**	965.71	$1^-, 2^-$
	20	100.00%	0.00	20888.53	5343.92	$2^+, 3^+$	100.00%	0.00	15007.67	3265.35	$1^-, 3^+$	100.00%	0.00	**11821.93**	3388.06	$1^-, 2^-$
	50	100.00%	0.00	93280.70	17697.95	$2^+, 3^+$	100.00%	0.00	67817.73	13010.55	$1^-, 3^*$	100.00%	0.00	**64425.13**	9267.66	$1^-, 2^*$

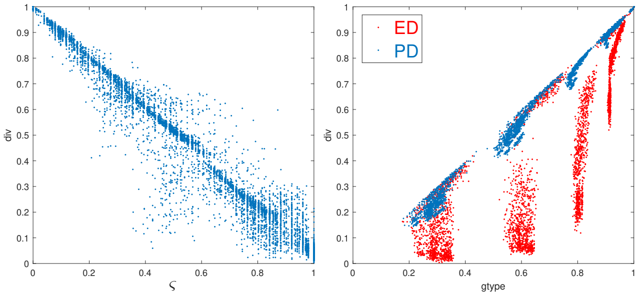
Table 2: Comparison in terms of diversity (gtype), the number of iterations (#iter) and results of statistical testing (stat) between variants with PD approach in unconstrained cases. Tests and notations used are the same as in Table 1.

n	μ	2-OPT(1)					3-OPT(2)					4-OPT(3)				
		gtype	**std**	**# iter**	**std**	**stat**	**gtype**	**std**	**# iter**	**std**	**stat**	**gtype**	**std**	**# iter**	**std**	**stat**
50	3	100.00%	0.00	83.43	42.67	$2^*, 3^+$	100.00%	0.00	67.03	31.22	$1^*, 3^*$	100.00%	0.00	**59.53**	39.87	$1^-, 2^*$
	10	100.00%	0.00	**1493.13**	380.08	$2^*, 3^*$	100.00%	0.00	1605.20	472.96	$1^*, 3^*$	100.00%	0.00	1748.20	659.40	$1^*, 2^*$
	20	100.00%	0.00	**26794.30**	7051.89	$2^-, 3^-$	99.96%	0.02	49955.43	244.10	$1^+, 3^*$	99.92%	0.02	50000.00	0.00	$1^+, 2^*$
	50	99.89%	0.01	**125000.00**	0.00	$2^*, 3^*$	99.76%	0.01	**125000.00**	0.00	$1^*, 3^*$	99.72%	0.01	**125000.00**	0.00	$1^*, 2^*$
100	3	100.00%	0.00	211.83	83.51	$2^+, 3^+$	100.00%	0.00	109.27	61.33	$1^-, 3^*$	100.00%	0.00	**107.07**	54.83	$1^-, 2^*$
	10	100.00%	0.00	1853.67	411.67	$2^+, 3^+$	100.00%	0.00	1563.40	412.61	$1^-, 3^*$	100.00%	0.00	**1403.73**	458.76	$1^-, 2^*$
	20	100.00%	0.00	**8377.83**	1259.16	$2^*, 3^-$	100.00%	0.00	9648.03	2079.70	$1^*, 3^*$	100.00%	0.00	9895.17	2165.04	$1^+, 2^*$
	50	99.94%	0.00	**500000.00**	0.00	$2^*, 3^*$	99.83%	0.01	**500000.00**	0.00	$1^*, 3^*$	99.77%	0.01	**500000.00**	0.00	$1^*, 2^*$
200	3	100.00%	0.00	451.60	249.54	$2^+, 3^+$	100.00%	0.00	285.97	155.56	$1^-, 3^*$	100.00%	0.00	**209.80**	79.16	$1^-, 2^*$
	10	100.00%	0.00	3161.17	751.84	$2^+, 3^+$	100.00%	0.00	2271.67	624.29	$1^-, 3^+$	100.00%	0.00	**1769.40**	384.22	$1^-, 2^-$
	20	100.00%	0.00	9776.93	1925.45	$2^+, 3^+$	100.00%	0.00	7550.90	1390.00	$1^-, 3^*$	100.00%	0.00	**7371.20**	1388.57	$1^-, 2^*$
	50	100.00%	0.00	**77706.33**	13011.40	$2^-, 3^-$	100.00%	0.00	106661.57	19402.75	$1^+, 3^-$	100.00%	0.00	132255.53	30508.53	$1^+, 2^+$
500	3	100.00%	0.00	828.90	320.42	$2^+, 3^+$	100.00%	0.00	552.70	276.51	$1^-, 3^*$	100.00%	0.00	**499.20**	258.36	$1^-, 2^*$
	10	100.00%	0.00	7036.17	1475.13	$2^+, 3^+$	100.00%	0.00	4686.20	972.09	$1^-, 3^*$	100.00%	0.00	**3806.03**	971.33	$1^-, 2^*$
	20	100.00%	0.00	19258.20	4942.90	$2^+, 3^+$	100.00%	0.00	13667.33	3053.62	$1^-, 3^*$	100.00%	0.00	**12294.77**	2767.25	$1^-, 2^*$
	50	100.00%	0.00	80004.13	9644.84	$2^+, 3^+$	100.00%	0.00	69792.40	13370.98	$1^-, 3^*$	100.00%	0.00	**63174.17**	10205.59	$1^-, 2^*$

limits. This suggests that when close to the optima, the probability of increasing *gtype* in an iteration decreases substantially with increasing μ/n. Also, in such cases, all variants with 2-OPT mutation operator achieve higher *gtype* score than those with 3-OPT, which in turn outperform those with 4-OPT. Furthermore, for $(n, \mu) = (50, 20)$, all 2-OPT variants always reaches the optimum, while 3-OPT variants struggle and 4-OPT variants fail entirely. This indicates that large-step mutation operators are prone to being stuck in local optima when the population is close to maximum *gtype*.

Additionally, minimising $\mathcal{N}(P)$ and minimising $\mathcal{D}(P)$ seem to produce the similar final *gtype* scores within similar numbers of iterations in all cases, given the same mutation operator used. This indicates that minimising $\mathcal{D}(P)$ also tends to maximise *gtype* when all tours are accepted. However, in the hard cases, the PD approach achieves somewhat lower *gtype* scores than ED across all variants. This suggests that the trade-offs between edge diversity and edge distances uniformity are non-trivial near optima.

Another observation is that with small enough μ/n, 2-OPT variants tend to take more iterations than 3-OPT variants, which in turn tend to terminate later than 4-OPT variants. When μ/n is larger



(a) ς and div

(b) $gtype$ and div

Figure 1: Scatter plots of all 8640 runs on the TSPlib instances. Each point corresponds to a final population after one run. The Pearson correlation coefficient between ς and div is -0.9815 with $p < 0.0001$.

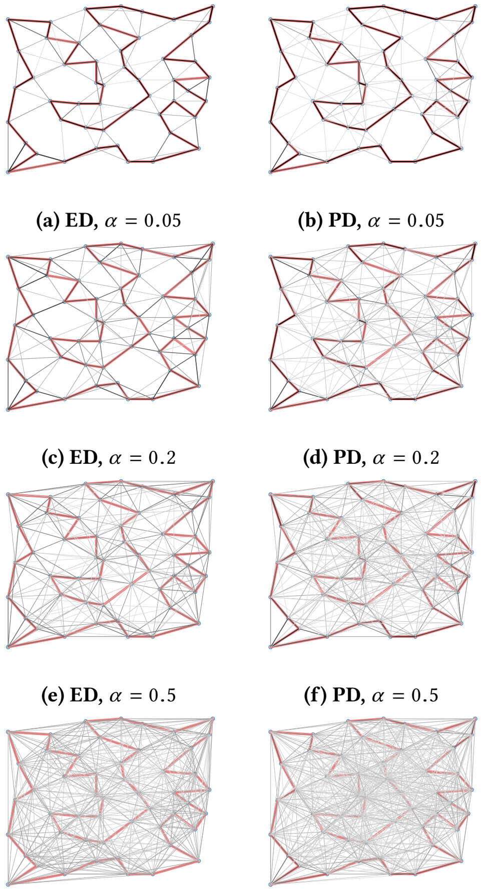
than some number, the trend seems to revert. However, the indication for this observation is weak since no statistically significant difference can be seen in many cases.

5.2 Constrained Diversity Optimisation

Now we consider TSP instances from the famous TSPlib, specifically eil51, eil76, eil101. For these experiments, we use the provided optimal tour for each instance, initialise P with μ copies of it and perform diversity maximisation subject to $c(I) \leq (1+\alpha)\cdot OPT$ for all $I \in P$. I. e., we accept tours only if they deviate in length by a factor of at most $(1+\alpha)$ from the optimal tour length OPT. We set up the instances for our experiments with $\mu = \{5, 10, 20, 50\}$ and $\alpha = \{0.05, 0.2, 0.5, 1.0\}$. For each instance, we run each algorithm variant 30 times and record the final population. As before, $gtype$ scores are reported in percentages, since these instances involve complete graphs and Theorem 1 applies. In addition, we record $\varsigma(P) = \left(\max_{X,Y\in P}\{o_{XY}\} - \min_{X,Y\in P}\{o_{XY}\}\right)/n$ to observe uniformity in edge distances between tours; lower scores indicate higher uniformity. All scores are averaged over 30 runs.

According to Table 3, $gtype$ scores achieved predictably increase with increasing α, albeit with diminishing return. On the other hand, ς scores tend to increase with increasing μ, and dramatically decrease with increasing α. In terms of mutation operators, 2-OPT seems to be the best performer overall, while 3-OPT and 4-OPT perform similar. Furthermore, all things equal, minimising $\mathcal{N}(P)$ tends to produce slightly higher $gtype$ scores and noticeably lower ς scores than minimising $\mathcal{D}(P)$. The PD approach seems to better capitalise on increasing α, improving edge distances uniformity faster. Moreover, the edge distances uniformity of the PD approach's output populations is less susceptible to compromise due to increasing μ within this range. These phenomena align with the remark that the ED approach focuses on $gtype$ while the PD approach compromises it for edge distances uniformity.

Additionally, we investigate the correlation between edge distances uniformity and the diversity score $div(P)$. Figure 1a shows a strong negative linear correlation between $div(P)$ and $\varsigma(P)$ across all cases. This suggests that focusing on edge distances uniformity, while maximising $gtype$, is effective in maximising div. Combined with earlier observations, we can conclude that the PD approach is much more likely to make a better compromise between maximising $gtype$ and maximising div. This is illustrated in Figure 1b, indicating that for each α value, the PD approach maintains higher div scores across all μ values without significantly sacrificing $gtype$ scores.



(a) ED, $\alpha = 0.05$

(b) PD, $\alpha = 0.05$

(c) ED, $\alpha = 0.2$

(d) PD, $\alpha = 0.2$

(e) ED, $\alpha = 0.5$

(f) PD, $\alpha = 0.5$

(g) ED, $\alpha = 1.0$

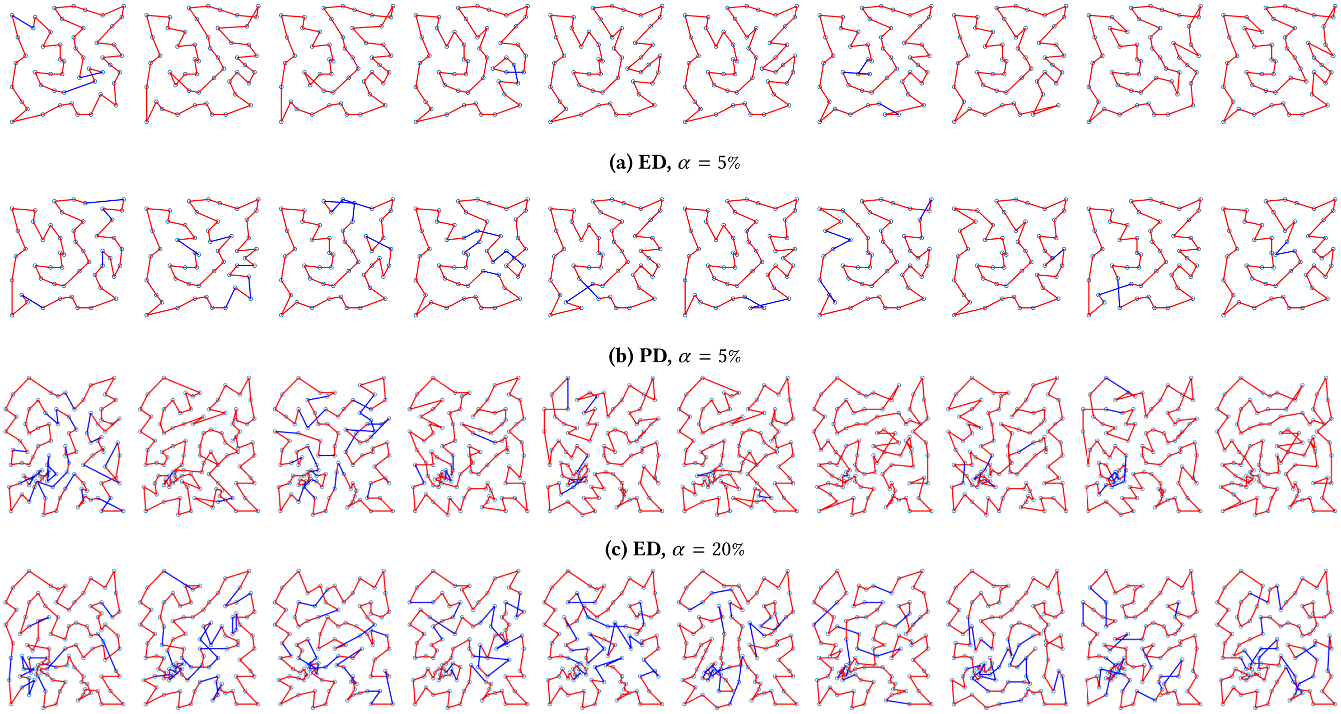
(h) PD, $\alpha = 1.0$

Figure 2: Visualised edge counts from resulted populations in eil51 cases with $\mu = 50$. The optimal tour is marked with red edges. Darker edges have higher counts.

The visuals in Figure 2 show that the PD approach results in fewer zero-count edges than the ED, regardless of α. It also results in higher maximum edge counts than the ED in those cases. This is because minimising $\mathcal{N}(P)$ flattens the edge counts distribution from the top down with the descending sorting order. The implication is that the counts distribution among the lower end is not guaranteed improvement, especially among the zero-count. On the other hand, minimising $\mathcal{D}(P)$, while not directly dealing with edge counts, tends to equalise this distribution while relaxing minimising higher count edges. Consequently, higher maximum edge counts are achieved, but fewer edges have high counts and more edges have low nonzero counts. Consequently, individuals are more likely to contain more unique edges (edges with count 1), which is also a mark

Table 3: Comparison in terms of diversity (gtype) and pairwise edge distances ranges (ς) among all variants of the EA on TSPlib instances. Better values with statistical significance (based on Wilcoxon rank sum tests with 95% confidence threshold) between ED and PD are bold-faced.

	μ	α	ED						PD					
			2-OPT		3-OPT		4-OPT		2-OPT		3-OPT		4-OPT	
			gtype	ς	gtype	ς	gtype	ς	gtype	ς	gtype	ς	gtype	ς
eil51	3	0.05	34.27%	68.37%	36.23%	67.78%	28.95%	74.05%	32.07%	69.54%	35.95%	66.14%	29.78%	71.96%
		0.2	70.78%	31.83%	67.93%	35.03%	63.68%	40.46%	71.11%	30.20%	65.95%	35.75%	63.12%	38.56%
		0.5	93.83%	8.43%	90.85%	11.05%	90.78%	11.37%	93.62%	**7.12%**	90.26%	11.18%	90.37%	10.59%
		1	99.89%	0.13%	99.83%	0.26%	99.80%	0.33%	99.80%	0.33%	99.74%	0.65%	99.76%	0.52%
	10	0.05	**31.82%**	79.67%	**33.64%**	78.43%	27.79%	82.88%	29.37%	78.24%	31.60%	**75.95%**	27.65%	**79.61%**
		0.2	**63.04%**	70.39%	**60.60%**	67.12%	**57.78%**	63.40%	60.99%	**44.90%**	59.23%	**47.39%**	55.99%	**51.31%**
		0.5	**83.14%**	30.85%	**81.34%**	36.67%	**81.29%**	37.58%	82.00%	**20.33%**	80.12%	**23.01%**	79.87%	**23.99%**
		1	**95.06%**	10.26%	**94.14%**	11.76%	**94.19%**	11.63%	94.57%	**7.58%**	93.30%	**8.50%**	93.16%	**8.50%**
	20	0.05	**32.52%**	95.23%	**32.86%**	91.37%	**27.89%**	90.85%	29.20%	**82.42%**	31.23%	**79.08%**	26.45%	**83.53%**
		0.2	**62.32%**	92.09%	**59.98%**	91.57%	**57.46%**	91.70%	59.30%	**49.15%**	57.84%	**51.37%**	54.53%	**55.29%**
		0.5	**80.95%**	59.02%	**79.24%**	63.27%	**79.60%**	64.84%	79.03%	**25.29%**	77.69%	**27.84%**	77.32%	**29.08%**
		1	**92.17%**	19.87%	**91.41%**	22.42%	**91.58%**	21.31%	90.97%	**11.83%**	90.28%	**12.42%**	90.24%	**12.75%**
	50	0.05	**32.66%**	100.00%	**33.23%**	100.00%	**28.01%**	100.00%	29.81%	**85.88%**	31.75%	**82.48%**	26.93%	**85.95%**
		0.2	**63.26%**	99.48%	**61.52%**	97.78%	**58.92%**	96.99%	59.68%	**53.59%**	58.36%	**57.12%**	55.04%	**59.80%**
		0.5	**80.74%**	92.35%	**79.56%**	90.59%	**79.80%**	88.76%	79.15%	**29.67%**	78.12%	**32.29%**	77.94%	**33.66%**
		1	**91.65%**	48.95%	**91.25%**	52.68%	**91.33%**	48.37%	90.60%	**15.69%**	90.42%	**15.75%**	90.42%	**16.67%**
eil76	3	0.05	30.12%	71.84%	30.06%	72.89%	24.20%	78.60%	29.04%	71.97%	29.80%	71.89%	24.08%	76.97%
		0.2	70.23%	31.67%	63.98%	38.90%	60.39%	42.02%	69.05%	32.15%	63.17%	37.76%	59.11%	42.06%
		0.5	94.66%	6.49%	90.63%	11.45%	91.05%	11.14%	94.72%	5.88%	90.82%	**9.74%**	90.29%	10.61%
		1	99.91%	0.22%	99.63%	0.57%	99.80%	0.48%	99.90%	0.09%	99.65%	0.53%	99.69%	0.57%
	10	0.05	**28.52%**	79.39%	**29.33%**	78.95%	23.14%	83.60%	27.09%	78.73%	28.35%	79.17%	22.87%	82.59%
		0.2	**61.56%**	57.19%	**58.62%**	56.75%	**56.01%**	60.13%	59.57%	**44.91%**	56.41%	**49.08%**	53.67%	**52.24%**
		0.5	**84.71%**	28.60%	**81.03%**	36.45%	**81.92%**	33.64%	82.94%	**18.99%**	79.54%	**23.07%**	79.45%	**23.90%**
		1	**95.92%**	7.94%	**94.39%**	9.96%	**94.78%**	9.43%	95.32%	**5.83%**	93.66%	**7.72%**	93.74%	**7.72%**
	20	0.05	**28.74%**	88.95%	**28.85%**	88.68%	**23.69%**	88.42%	26.96%	**81.49%**	27.38%	**81.97%**	22.25%	**85.18%**
		0.2	**61.19%**	92.76%	**58.37%**	92.89%	**56.06%**	91.45%	58.50%	**48.29%**	55.41%	**53.16%**	52.20%	**56.67%**
		0.5	**82.35%**	53.90%	**79.64%**	65.39%	**80.32%**	58.68%	79.72%	**23.73%**	77.58%	**26.84%**	77.40%	**28.42%**
		1	**93.03%**	16.10%	**91.95%**	20.04%	**92.37%**	18.11%	91.67%	**10.26%**	90.67%	**11.58%**	90.78%	**11.84%**
	50	0.05	**28.84%**	100.00%	**28.41%**	99.91%	**23.99%**	98.95%	27.01%	**84.47%**	27.77%	**85.09%**	22.52%	**87.28%**
		0.2	**62.42%**	98.95%	**59.32%**	98.73%	**56.80%**	98.07%	58.16%	**52.54%**	55.73%	**56.80%**	52.29%	**60.35%**
		0.5	**81.53%**	86.62%	**79.74%**	90.39%	**80.35%**	88.03%	79.05%	**27.50%**	77.21%	**31.01%**	77.08%	**32.63%**
		1	**91.73%**	42.19%	**91.18%**	46.62%	**91.39%**	38.55%	90.32%	**13.68%**	89.84%	**14.78%**	90.02%	**14.91%**
eil101	3	0.05	35.84%	65.97%	**36.79%**	65.74%	31.21%	70.99%	35.69%	65.71%	35.49%	65.51%	30.24%	70.69%
		0.2	**72.61%**	29.60%	67.50%	34.69%	65.30%	36.40%	71.29%	29.11%	65.56%	35.08%	63.99%	36.83%
		0.5	95.03%	5.91%	91.86%	9.64%	92.29%	9.27%	94.72%	5.78%	91.60%	8.98%	91.63%	9.11%
		1	99.79%	0.40%	99.49%	0.76%	99.72%	0.50%	99.81%	0.36%	99.57%	0.66%	99.85%	**0.23%**
	10	0.05	31.88%	76.27%	**33.44%**	74.19%	**28.88%**	78.88%	31.36%	**73.37%**	32.60%	**72.05%**	27.77%	**76.93%**
		0.2	**63.70%**	49.64%	**60.78%**	54.16%	**58.69%**	55.38%	62.44%	**40.89%**	59.29%	**44.95%**	56.49%	**47.95%**
		0.5	**85.14%**	25.78%	**82.55%**	30.43%	**82.75%**	33.00%	83.84%	**17.52%**	81.64%	**20.20%**	80.78%	**21.55%**
		1	**96.29%**	6.80%	**95.07%**	9.08%	**95.25%**	8.75%	95.97%	**4.95%**	94.37%	**6.77%**	94.37%	**6.86%**
	20	0.05	**32.80%**	85.28%	**33.52%**	85.31%	**28.74%**	85.97%	30.58%	**76.63%**	31.78%	**75.38%**	27.21%	**79.90%**
		0.2	**63.37%**	92.48%	**60.27%**	89.70%	**58.75%**	89.90%	60.83%	**44.36%**	58.11%	**48.91%**	55.30%	**51.16%**
		0.5	**82.93%**	56.17%	**80.68%**	57.49%	**81.19%**	55.68%	81.04%	**21.22%**	79.08%	**24.19%**	79.05%	**25.21%**
		1	**93.71%**	14.98%	**92.62%**	17.99%	**93.06%**	15.91%	92.28%	**9.01%**	91.31%	**10.23%**	91.42%	**10.43%**
	50	0.05	**34.32%**	100.00%	**33.17%**	99.47%	**28.96%**	99.01%	30.26%	**79.44%**	31.34%	**78.75%**	26.91%	**82.71%**
		0.2	**63.92%**	98.48%	**60.95%**	97.92%	**59.39%**	97.66%	59.76%	**48.25%**	57.89%	**52.24%**	54.86%	**55.74%**
		0.5	**81.65%**	86.30%	**79.96%**	90.23%	**80.51%**	85.87%	79.34%	**24.85%**	78.20%	**27.36%**	77.85%	**28.84%**
		1	**91.58%**	34.69%	**90.88%**	45.38%	**91.20%**	36.60%	90.01%	**12.34%**	89.60%	**13.17%**	89.80%	**13.43%**



(a) ED, $\alpha = 5\%$

(b) PD, $\alpha = 5\%$

(c) ED, $\alpha = 20\%$

(d) PD, $\alpha = 20\%$

Figure 3: Visualised tour populations from resulted populations in eil51 cases with $\mu = 10$ and 2-OPT as mutation operator. Red edges are shared by at least two tours in the population, and blue ones are unique to the tour. Left-to-right is the ascending tour length order.

of highly diverse populations. As Figure 3 shows, with small α, the populations generated by the ED approach tend to contain duplicated tours and tours without unique edges. In contrast, tours produced by the PD approach tend to exhibit more uniqueness and stand out from the rest.

6 CONCLUSION

Evolutionary diversity optimisation aims to generate a set of diverse solutions where all solutions meet given quality criteria. We have introduced and examined for the first time evolutionary diversity optimisation for a classical combinatorial optimisation problems. We introduced two diversity measures that can be used for the Traveling Salesperson Problem and evaluated their performance when used for simple population-based elitist evolutionary algorithms. The results show that both measures can be optimized well in the unconstrained case where all tours meet the quality criterion. Furthermore, our investigations for TSPlib instances point out the increase in terms of diversity that can be obtained when relaxing the quality constraint determined by the required approximation ratio. We also highlighted some differences between populations generated by these two approaches.

The focus of this paper has been on the introduced diversity measures for diversity optimisation of the TSP and their performance in simple population-based elitist evolutionary algorithms. For future research, it would be interesting to incorporate these measures into state of the art evolutionary algorithms for the TSP and evaluate their performance.

ACKNOWLEDGMENT

This work has been supported by the Australian Research Council (ARC) through grant DP190103894.

REFERENCES

[1] Bradley Alexander, James Kortman, and Aneta Neumann. 2017. Evolution of artistic image variants through feature based diversity optimisation. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO*. ACM, 171–178. https://doi.org/10.1145/3071178.3071342

[2] Brian Alspach, Jean-Claude Bermond, and Dominique Sotteau. 1990. *Decomposition into cycles I: Hamilton decompositions*. Springer Netherlands, 9–18.

[3] Jakob Bossek, Pascal Kerschke, Aneta Neumann, Markus Wagner, Frank Neumann, and Heike Trautmann. 2019. Evolving diverse TSP instances by means of novel and creative mutation operators. In *Proceedings of the 15th ACM/SIGEVO Conference on Foundations of Genetic Algorithms, FOGA*. ACM, 58–71. https://doi.org/10.1145/3299904.3340307

[4] Gregory W. Corder and Dale I. Foreman. 2009. *Nonparametric Statistics for Non-Statisticians: A Step-by-Step Approach*. Wiley.

[5] Wanru Gao, Samadhi Nallaperuma, and Frank Neumann. 2016. Feature-based diversity optimization for problem instance classification. In *Parallel Problem Solving from Nature, PPSN (LNCS)*, Vol. 9921. Springer, 869–879. https://doi.org/10.1007/978-3-319-45823-6_81

[6] Keld Helsgaun. 2000. An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic. *European Journal of Operational Research (EJOR)* 126 (2000), 106–130. https://doi.org/10.1016/S0377-2217(99)00284-2

[7] Jorge Kanda, Andre Carvalho, Eduardo Hruschka, and Carlos Soares. 2011. Selection of Algorithms to Solve Traveling Salesman Problems Using Meta-Learning. *International Journal of Hybrid Intelligent Systems* 8, 3 (2011), 117–128.

https://doi.org/10.3233/HIS-2011-0133

[8] Pascal Kerschke, Holger H. Hoos, Frank Neumann, and Heike Trautmann. 2019. Automated Algorithm Selection: Survey and Perspectives. *Evolutionary Computation* 27, 1 (2019), 3–45. https://doi.org/10.1162/evco_a_00242

[9] Pascal Kerschke, Lars Kotthoff, Jakob Bossek, Holger H. Hoos, and Heike Trautmann. 2018. Leveraging TSP Solver Complementarity through Machine Learning. *Evolutionary Computation* 26, 4 (2018), 597–620. https://doi.org/10.1162/evco_a_00215

[10] Shen Lin and Brian Wilson Kernighan. 1973. An Effective Heuristic Algorithm for the Traveling-Salesman Problem. *Operations Research* 21, 2 (1973), 498–516. https://doi.org/10.1287/opre.21.2.498

[11] Olaf Mersmann, Bernd Bischl, Heike Trautmann, Markus Wagner, Jakob Bossek, and Frank Neumann. 2013. A Novel Feature-Based Approach to Characterize Algorithm Performance for the Traveling Salesperson Problem. *Annals of Mathematics and Artificial Intelligence* 69 (2013), 151–182. Issue 2. https://doi.org/10.1007/s10472-013-9341-2

[12] Yuichi Nagata and Shigenobu Kobayashi. 2013. A Powerful Genetic Algorithm Using Edge Assembly Crossover for the Traveling Salesman Problem. *INFORMS Journal on Computing* 25, 2 (2013), 346–363. https://doi.org/10.1287/ijoc.1120.0506

[13] Aneta Neumann, Wanru Gao, Carola Doerr, Frank Neumann, and Markus Wagner. 2018. Discrepancy-based evolutionary diversity optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO*. ACM, 991–998. https://doi.org/10.1145/3205455.3205532

[14] Aneta Neumann, Wanru Gao, Markus Wagner, and Frank Neumann. 2019. Evolutionary diversity optimization using multi-objective indicators. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO*. ACM, 837–845. https://doi.org/10.1145/3321707.3321796

[15] Gerhard Reinelt. 1991. TSPLIB - A Traveling Salesman Problem Library. *INFORMS Journal on Computing* 3, 4 (1991), 376–384.

[16] Kate Amanda Smith-Miles and Jano Ilja van Hemert. 2011. Discovering the Suitability of Optimisation Algorithms by Learning from Evolved Instances. *Annals of Mathematics and Artificial Intelligence* 61, 2 (2011), 87–104. https://doi.org/10.1007/s10472-011-9230-5

[17] Tamara Ulrich, Johannes Bader, and Eckart Zitzler. 2010. Integrating decision space diversity into hypervolume-based multiobjective search. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO*. ACM, 455–462.

[18] Tamara Ulrich and Lothar Thiele. 2011. Maximizing population diversity in single-objective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO*. ACM, 641–648.

[19] Handing Wang, Yaochu Jin, and Xin Yao. 2017. Diversity Assessment in Many-Objective Optimization. *IEEE Transactions on Cybernetics* 47, 6 (2017), 1510–1522. https://doi.org/10.1109/TCYB.2016.2550502

[20] Xiao-Feng Xie and Jiming Liu. 2009. Multiagent Optimization System for Solving the Traveling Salesman Problem (TSP). *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 39, 2 (2009), 489–502. https://doi.org/10.1109/TSMCB.2008.2006910

[21] Kenny Q. Zhu and Ziwei Liu. 2004. Population Diversity in Permutation-Based Genetic Algorithm. In *Machine Learning: ECML 2004*. Springer Berlin Heidelberg, 537–547.