



# A Comparison of Genetic Representations and Initialisation Methods for the Multi-objective Shortest Path Problem on Multigraphs

Lilla Beke<sup>1</sup> · Michal Weiszer<sup>1</sup> · Jun Chen<sup>1</sup>

Received: 8 August 2020 / Accepted: 8 February 2021 / Published online: 30 March 2021  
© The Author(s) 2021

## Abstract

This paper compares different solution approaches for the multi-objective shortest path problem (MSPP) on multigraphs. Multigraphs as a modelling tool are able to capture different available trade-offs between objectives for a given section of a route. For this reason, they are increasingly popular in modelling transportation problems with multiple conflicting objectives (e.g., travel time and fuel consumption), such as time-dependent vehicle routing, multi-modal transportation planning, energy-efficient driving, and airport operations. The multigraph MSPP is more complex than the NP-hard simple graph MSPP. Therefore, approximate solution methods are often needed to find a good approximation of the true Pareto front in a given time budget. Evolutionary algorithms have been successfully applied for the simple graph MSPP. However, there has been limited investigation of their applications to the multigraph MSPP. Here, we extend the most popular genetic representations to the multigraph case and compare the achieved solution qualities. Two heuristic initialisation methods are also considered to improve the convergence properties of the algorithms. The comparison is based on a diverse set of problem instances, including both bi-objective and triple objective problems. We found that the metaheuristic approach with heuristic initialisation provides good solutions in shorter running times compared to an exact algorithm. The representations were all found to be competitive. The results are encouraging for future application to the time-constrained multigraph MSPP.

**Keywords** Multi-objective shortest path problems · Multigraphs · Genetic representation techniques · Heuristic initialisation

## Introduction

There is substantial evidence [1–5] that modelling transportation problems as multigraphs offer benefits with regards to time, cost, environmental impact, and flexibility in multiple practical settings. The adoption of multiple parallel edges between pairs of nodes is rooted in the multi-objective nature of the problems. When there are multiple ways to traverse a section of the route, offering different trade-offs

between the objectives, they should all be modelled in the optimisation process to find a set of Pareto-optimal solutions. The parallel edges in practice might correspond to different physical routes as in multi-objective vehicle routing problems [2, 6], different modes of transport [3], or the same physical route traversed with different speed profiles such as in energy-efficient driving [4, 5] and in the airport ground movement problem [1].

The multi-objective shortest path problem (MSPP) on multigraphs can be decomposed to two intertwined NP-hard problems, the MSPP on simple graphs [7] and the fixed sequence arc selection problem (FSASP) [6]. In the FSASP, the sequence of nodes to be traversed is fixed, but there are multiple parallel edges between any neighbouring nodes in the sequence. Consequently, finding the exact solutions for a multigraph MSPP often requires an unacceptably long time, in particular, in real-world applications.

For the simple graph MSPP, metaheuristics and, in particular, genetic algorithms have been used with success [8–10] to obtain a good representation of the Pareto-optimal

---

This article is part of the topical collection “Evolution, the New AI Revolution” guest edited by Anikó Ekárt and Anna Isabel Esparcia-Alcázar.

---

✉ Lilla Beke  
l.beke@qmul.ac.uk  
Jun Chen  
jun.chen@qmul.ac.uk

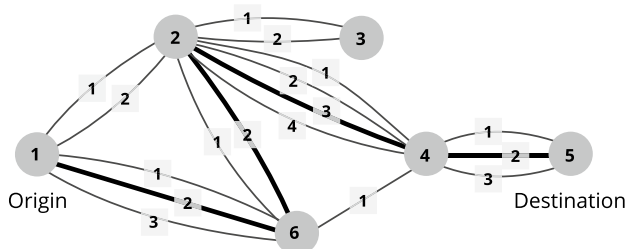
<sup>1</sup> School of Engineering and Materials Science, Queen Mary of London, London, UK

solutions in a short time. Genetic algorithms imitate the biological process of natural selection to find near-optimal solutions to computationally difficult problems. A key aspect of this process is the chosen representation method, that determines how the candidate solutions are encoded to the so-called chromosomes for the evolutionary process. The representation method has an important role in determining the search landscape and thus affects the effectiveness of the search [11]. Moreover, the representation also determines the kind of crossover and mutation operators that can be used.

Several different representation methods and genetic operators have been proposed for the simple graph MSPP. The multigraph MSPP, on the other hand, received considerably less attention, even though it is a relevant issue for a wide range of real-world problems and a natural extension to the body of research investigating the simple graph MSPP. To our knowledge, the only previous attempts at applying metaheuristics to the multigraph MSPP were in the context of multi-modal transportation [12, 13]. Both of these studies used genetic algorithms and extend the direct variable length [14] representation. There are three other representations proposed for the simple graph shortest path problems, the direct fixed length [15], random key [16], and integer-valued priority [10] representations. In this paper, we investigate and extend all the four mentioned representations.

The representations introduced for the simple graph problem cannot be used without modification for the multigraph MSPP. The extension requires the incorporation of the choice between parallel edges into the chromosomes. Figure 1 illustrates an example of a solution path in a multigraph.

Heuristic initialisation techniques have been proven to be useful in different combinatorial optimisation problems [17–20], as they can lead to quicker convergence by starting the evolutionary process with an already high-quality initial population. This is done by locating promising areas of the search space through utilising a priori information about the specific problem. The initial population is then drawn partially or exclusively from those promising areas.



**Fig. 1** An example of a multigraph network and a solution path from node 1 to node 5 in it, indicated by the bold lines

A common concern about such initialisation methods is that they might lead to premature convergence caused by a lack of sufficient diversity. This explains why there are relatively few studies utilising heuristic initialisation for the MSPP. The problem of premature convergence can be mitigated by introducing enough randomness into the population, while still preserving higher quality compared to a purely random population.

We employ a heuristic initialisation method utilising single objective search. This method was introduced for the case of direct representation in [9], and here, we extend it to the case of the other representations. Moreover, we also introduce a novel heuristic initialisation method based on the idea of discouraging detours, that is applicable to all the representations used.

## Contributions

The main contributions of this paper:

- Four representations that were originally proposed for simple graph shortest path problems are adapted and extended to the multigraph MSPP. Three of these representations have not been previously applied to the multigraph MSPP.
- A novel indirect method of incorporating the choice of parallel edges into the chromosomes is proposed. This method makes it possible to adopt the priority-based representations to the multigraph problem with inhomogeneous numbers of parallel edges without the use of any repairing mechanism. The method can also be used with the direct representations.
- The direct [12] and indirect ways of encoding parallel edges are compared for the direct variable length representation.
- A heuristic initialisation method based on graph structure is proposed and shown to provide additional benefits when used together with the existing heuristic initialisation method [9].
- The representation and initialisation methods are compared on a diverse problem set using dominance-compliant quality indicators. It is observed that some representations are better suited for specific graph types.

This study extends our previous work [21]. The first main addition compared to [21] is the inclusion of more variants of different representations. Variants include (1) different crossover operators, (2) direct way of encoding parallel edges for the direct variable length representations, and (3) adapting an existing heuristic initialisation method [9] to priority-based representations. Second, a larger set of problem instances is used for the comparison, and the performance

of the exact algorithm on these instances is described in more detail.

## Structure of Paper

In the rest of the paper, “” describes the multigraph MSPP problem. Related work on representation methods for the simple and multigraph MSPP, and initialisation techniques are summarised in “[Related Work](#)”. “[Proposed Representations and Initialisation Methods](#)” describes our approach to extend the main representations to the multigraph case, the genetic operators used, and the proposed heuristic initialisation technique. “[Details](#)” contains the implementation details of the algorithms, description of test instances, and the evaluation of the approximate solution sets. In “[Results](#)”, numerical experiments and their results are presented. Finally, conclusion is drawn and future research directions are discussed in “[Conclusion and Future Work](#)”.

## Problem Description

The multigraph MSPP is defined by a multigraph network  $G = (V, E)$  and a multi-dimensional cost-vector associated with each edge in  $G$ . The network is assumed to be undirected in this paper.  $V = 1, \dots, n$  represents the set of nodes, and  $E$  represents the set of edges. Given that  $G$  is a multigraph, there might be multiple edges in  $E$  connecting the same nodes. Thus, an edge in the network is denoted by  $e = (u, v, i)$ , where  $u, v \in V$  and  $i$  is a parallel edge index, which differentiates between the edges with the same endpoints  $u$  and  $v$ . These edges are numbered as  $i(u, v)$ , where  $i$  is starting from 1 up to  $l(u, v)$ , where  $l(u, v)$  is the number of parallel edges between the two nodes  $u$  and  $v$ . There are two special nodes  $v_O, v_D \in V$ , the origin and destination nodes, respectively. The costs associated with each edge according to  $k$  objectives considered are given by a  $k$ -dimensional cost-vector  $\text{cost}(e) = (c_1(e), c_2(e), \dots, c_k(e))$ . For a valid path  $P$  between  $v_O$  and  $v_D$ , the corresponding cost-vector can be calculated according to Eq. (1).

$$C(P) = \sum_{e \in P} \overline{\text{cost}(e)}. \quad (1)$$

We are looking for the minimum cost path between  $v_O$  and  $v_D$  considering all  $k$  objectives. The solution is a set of valid paths with non-dominated cost-vectors, i.e., the Pareto-optimal solutions. A solution path  $P_1$  is said to be Pareto-optimal if there is no solution path that is at least as good as  $P_1$  according to all  $k$  objectives and better according to at least one objective.

## Related Work

In this section, we review the genetic representation methods and operators proposed for simple graph shortest path problems and the multigraph MSPP. Initialisation techniques are also discussed.

### Genetic Representations for Paths in Simple Graphs

In the single objective shortest path problem (SSPP) and simple graph MSPP, candidate solutions need to encode a sequence of neighbouring nodes in a graph. Several genetic representation methods have been introduced for these problems. These methods can be classified as direct representations and priority-based representations.

#### Direct Variable Length Encoding

The most straightforward way of encoding a path in a graph is the direct variable length representation proposed in [22] for the SSPP. Chromosomes consist of lists of node IDs, that form a path starting with the origin node. An arbitrary list of nodes usually will not correspond to a feasible path in the graph, and this necessitates the use of problem specific genetic operators, which maintain feasibility of the path. Ahn and Ramakrishna introduced a crossover based on common nodes in the solution paths and a mutation based on a random walk for the SSPP [14], which were adapted by multiple authors for the multi-objective problem [8, 9, 23].

The main advantage of this representation is that it gives a one-to-one mapping, which is usually preferable over one-to-n mapping, since it avoids introducing plateaus in the search space. In one-to-n mapping, several different chromosomes might encode the same solution path, and thus have the same associated fitnesses, forming a plateau. The algorithm then might rediscover the same solutions over and over again through different chromosomes and waste computational resources.

A disadvantage is that the genetic operators might lead to loop formation, and thus offspring need to be checked and repaired after mutation and crossover. Also, according to [24, 25], this representation is not suitable for large networks.

An example of a chromosome with direct variable length representation that encodes the solution path in Fig. 1 without specifying the choice between parallel edges is:

[1, 6, 2, 4].

#### Direct Fixed Length Encoding

Another node ID-based representation was proposed by Inagaki et al. [15]. The length of the chromosome equals  $n$ , the

number of nodes in the network, and the node IDs are the numbers from 1 to  $n$ . The chromosomes incorporate a pointer to a neighbouring node for each node. The locus of a gene corresponds to a node in the network with the same ID, and the value of the gene is the ID of a neighbour that should be the next node in the path. The path is decoded by following the pointers until the destination is reached or a loop is formed.

Note that this is an one-to- $n$  representation, because, generally, there are genes whose values do not affect the decoding process. The crossover operator applied in Ref. [15] is essentially uniform crossover. This approach is deemed ineffective and requires large population sizes [14, 24].

An example of a chromosome with direct fixed length representation that encodes the solution path in Fig. 1 without specifying the choice between parallel edges is:

[6, 4, 2, 5, 4, 2].

### Integer-Valued Priority-Based Encoding

Gen et al. proposed a priority-based encoding technique [26] that uses integer priority values to encode solution paths indirectly. Chromosomes consist of some permutation of the integers from 1 to  $n$ . The priority assigned to a node with ID  $i$  is given by the value of a gene at locus  $i$ .

A path is decoded from the chromosome by starting at the origin node and step-by-step moving to the neighbouring node with the highest priority, given it is not yet in the path. If it is already in the path, the neighbour with the next highest priority is chosen instead, if there is one.

The main advantage of this representation is that a random permutation of the priorities will always be decoded to some valid path starting from the origin node. This means that more traditional crossover operators can be used and expected to produce feasible paths, unlike in the case of the direct representations.

In Ref. [26], position-based crossover (PX) was used. An extension of the one-point crossover for integer-valued priority representation, the weight mapping crossover (WMX), was proposed specifically for this problem [10]. This approach has been shown to perform well in comparison to the direct variable length representation for the bi-objective problem.

An example of a chromosome with integer-valued priority representation that encodes the solution path in Fig. 1 without specifying the choice between parallel edges is:

[6, 3, 1, 2, 5, 4].

### Random Key-Based Encoding

Gen and Lin proposed another similar encoding technique using floating-point numbers instead of integers as priorities

[27], resulting in a random key representation for the SSPP. Random keys were found to be a powerful method for permutation representation in other combinatorial optimisation problems [28].

The advantage of random key encoding is that we do not have to maintain distinct priority values, as most of the time, the value of two priorities compared will not be equal. This makes it possible to use different crossover operators that are not designed specifically for the MSPP, such as arithmetical crossover, uniform crossover, or two-point crossover. In Ref. [27], arithmetical crossover is employed, where the offspring are calculated as the weighted average of the two parent chromosomes. The authors report higher search capability, enhanced rate of reaching optimal solutions, and improved computation time compared to the integer-valued priority-based encoding and the direct variable length encoding.

An example of a chromosome with random key representation that encodes the solution path in Fig. 1 without specifying the choice between parallel edges is:

[0.93, 0.36, 0.12, 0.25, 0.51, 0.45].

### Genetic Representations for Paths in Multigraphs

To encode solution paths in multigraphs, the indices of parallel edges used in the path also need to be specified.

Abbaspour and Samadzadegan extended the direct variable length representation to the multi-modal transportation problem [12] by indicating the mode of travel for each edge in the path. The chromosome is twice as long as the number of nodes in the network. The genes at odd loci correspond to node IDs, as in Ref. [14]. The genes at even loci are used to indicate the mode of travel between the nodes.

Yu and Lu [13] proposed a slightly different approach, where they only indicated the changes in the mode of travel and do not specify it for each edge separately. Genes indicating the mode of travel for the consecutive node IDs have a negative value, to differentiate them from the node IDs. However, only a limited number of parallel edges were used. When the number of parallel edges grows, the possible advantage of a shorter chromosome is diminished by the burden of maintaining the chromosome structure with more complicated operators.

### Initialisation

Most mentioned works applied random initialisation, which implies a random walk starting from the origin node, random pointers to neighbours, or random priorities. There are two notable examples when heuristic initialisation methods were used, both with the direct variable length representation. (1) Information about the spatial location of the nodes in the

graph is utilised [23]. The initial solutions are generated in a way that each edge moves away from the origin node and closer to the destination node measured by Euclidean distance. However, spatial information is not always available and in some cases might be misleading. (2) Single objective search is utilised [9]. Shortest paths are found according to weighted aggregations of smaller subsets of the objectives with the Dijkstra's algorithm. These are included in the initial population along with randomly generated chromosomes.

A heuristic initialisation approach was proposed for solving the SSPP with particle swarm optimisation and priority-based representation in Ref. [25]. Similar to [23], this method aims to decrease the risk of loop formation, but it uses information about the structure of the graph instead of spatial information. Priorities were randomly assigned and node IDs were assigned in a way that the higher the IDs are, the closer a node is to the destination. Then, detours in solution paths can be detected through finding decreasing node IDs. Not all detours were prohibited, only the ones where the decrease in node IDs is above some pre-specified limit.

## Proposed Representations and Initialisation Methods

### Encoding Parallel Edges

Representations for the multigraph MSPP need to encode the parallel indices for each pair of consecutive nodes in a solution path, not just the node sequence. The previously proposed methods for encoding mode of transport in the multi-modal transportation literature employ the direct variable length representation, which allows for direct inclusion of the parallel edge indices. Extending the priority-based representations to the multigraph case is less straightforward.

The multigraph might contain varied numbers of parallel edges between pairs of nodes, as in the multi-modal transportation problem. This does not cause a problem in the direct representations, because once the parallel edge indices are initialised to be feasible, and the crossover and mutation operators do not ruin feasibility. This is not the case with the priority-based representations.

Following a similar strategy for the priority-based representations to assign parallel edge indices to node IDs may lead to infeasible solutions after genetic operators. Take an example of a crossover that results in a solution path in which a node  $u$  is followed by another node  $v$ . In both of the parent solution paths, the node  $u$  might be followed by nodes other than  $v$ . Then, the parallel indices assigned to node  $u$  in the parents might be higher than what is available between  $u$  and  $v$ .

One possible way of overcoming this problem would be to apply a repairing mechanism after the solution path is decoded. Apparently, the choice between parallel edges would be mostly governed by this repairing mechanism, not taking full advantage of the exploration and exploitation capabilities rendered by the mechanism of the evolutionary process. Also, the computational burden would increase. Here, we propose an alternative method for representing the parallel indices that does not require repairing.

Instead of encoding the indices of parallel edges directly, we use a floating-point number  $r$  between 0 and 1, denoted the parallel edge indicator. Parallel edge indicators are assigned to node IDs, and they encode which parallel edge to use when leaving the given node. Given two neighbouring nodes  $u, v$  and the number of parallel edges between them  $l(u, v)$ , the index of the chosen parallel edge can be calculated as  $\lceil r(u) * l(u, v) \rceil + 1$  when moving from  $u$  to  $v$  (indexing starts at 1). Any random value of parallel edge indicators can be decoded to an index of a parallel edge that is available between two given nodes. The parallel edge indicators are employed with all four representations as a way of extending them to the multigraph problem.

One possible drawback is the increased number of alternative chromosomes that can encode the same solution path, leading to the increased ambiguity of the representations. To investigate the effect of this ambiguity, the directly encoded parallel edge indices [12] are also implemented with the direct variable length representation. This provides a way of encoding solution paths in the multigraph without any ambiguity, providing an interesting case for comparison with all other representations. The directly encoded parallel edge indices are also applicable to the direct fixed length representation also. We did not include this in our investigation, because that representation already includes some ambiguity, as noted in “[Direct Fixed Length Encoding](#)”.

The directly encoded parallel edge indices are also represented as floating-point numbers assigned to node IDs. The value of a parallel edge index is divided by 100 to get a floating-point number. Therefore, up to 100 parallel edges can be represented between two nodes.

### Direct Variable Length Representation for the Multigraph MSPP

We implement two versions of this representation, abbreviated *DirVarL-indir* and *DirVarL-dir*. The difference is in the encoding of the choice between parallel edges, which is done indirectly with parallel edge indicators for *DirVarL-indir*, and for *DirVarL-dir*, it is done directly with parallel edge indices. Note that *DirVarL-dir* is similar to the solution representation used in Ref. [12], but we use different genetic operators. [12] employed an expensive mutation operator that includes using the Dijkstra's algorithm and a repairing



mechanism. Instead, we use the mutation suggested in [14], as it is most often used with the direct variable length representation.

The base of the chromosome is a sequence of node IDs, that are integers, such as in [14]. As the parallel edge indicators are floating-point numbers below 1, it is convenient to add them to the node IDs. This way, the chromosome is the same length as in the simple graph problem. The integer parts indicate the node sequence of the path and the fractional parts indicate which of the parallel edges to use when leaving the nodes.

The solution path can be decoded according to Algorithm 1. The algorithm loops through the chromosome (lines 3–13) and collects the nodes and indices determining the solution path. If the choice of parallel edges is encoded directly, lines 8–9 are executed, and if they are encoded indirectly, line 11–12 is used to decode the parallel indices.

**Algorithm 1:** Decode direct variable length chromosome for the multigraph MSPP

```

Input: chromosome,  $G, v_O, v_D, useDirectIndices$ 
Output: nodes: node sequence, indices: sequence of parallel indices
1 nodes  $\leftarrow$  List with a single element:  $v_O$ 
2 indices  $\leftarrow$  Empty list
3 for  $i \leftarrow 2$  to length of chromosome do
4   prevNode  $\leftarrow$  Integer part of the  $(i - 1)$ th gene of chromosome
5   nextNode  $\leftarrow$  Integer part of the  $i$ th gene of chromosome
6   Add nextNode to nodes
7   if useDirectIndices is True then
8     frac  $\leftarrow$  fractional part of the  $(i - 1)$ th gene of chromosome
9     currentIndex  $\leftarrow 100 * frac$ 
10  else
11     $r \leftarrow$  fractional part of the  $(i - 1)$ th gene of chromosome
12    currentIndex  $\leftarrow \lfloor r * l(prevNode, nextNode) \rfloor + 1$ 
13  Add currentIndex to indices
14 return nodes, indices
    
```

An example of a chromosome with *DirVarL-indir* that encodes the solution path in Fig. 1 is:

[1.38, 6.82, 2.67, 4.51],

where the value of the first gene encodes that the first node in the solution path is 1 and the parallel edge to be used to reach node 6 is calculated as  $\lfloor r(1) * l(1, 6) \rfloor + 1 = \lfloor 0.38 * 3 \rfloor + 1 = 2$ . An example of a chromosome with *DirVarL-dir* that encodes the solution path in Fig. 1 is:

[1.02, 6.02, 2.03, 4.02].

**Direct Fixed Length Representation for the Multigraph MSPP**

The parallel edge indicators are added to the genes to extend the representation from [15]. The solution path can be decoded according to Algorithm 2. The loop in lines 3–12 is executed until the destination node is reached by the path, or a node would appear a second time in the path. The next node to add to the path is found in line 5, and if it is not yet in the node sequence, the parallel index to use is decoded in lines 10–12 and added to the parallel index sequence.

**Algorithm 2:** Decode direct fixed length chromosome for the multigraph MSPP

```

Input: chromosome,  $G, v_O, v_D$ 
Output: nodes: node sequence, indices: sequence of parallel indices
1 nodes  $\leftarrow$  List with a single element:  $v_O$ 
2 indices  $\leftarrow$  Empty list
3 while Last node in nodes  $\neq v_D$  do
4   prevNode  $\leftarrow$  Last element of nodes
5   nextNode  $\leftarrow$  Integer part of the prevNode-th element of chromosome
6   if nextNode is in nodes then
7     Add  $v_D$  to nodes // Path got stuck, will be penalised
8   else
9     Add nextNode to nodes
10     $r \leftarrow$  fractional part of prevNode-th element of chromosome
11    currentIndex  $\leftarrow \lfloor r * l(prevNode, nextNode) \rfloor + 1$ 
12    Add currentIndex to indices
13 return nodes, indices
    
```

We abbreviate the direct fixed length representation for the multigraph MSPP using parallel edge indicators *DirFixL*.

An example of a chromosome with *DirFixL* that encodes the solution path in Fig. 1 is:

[6.38, 4.67, 2.24, 5.51, 4.09, 2.82].

**Integer-Valued Priority-Based Representation for the Multigraph MSPP**

The parallel edge indicators are added to the genes to extend the representation from [10]. Then, the priority value of node  $v$  can be found as the integer part of the  $v$ th gene in the chromosome, and the parallel edge indicator for node  $v$  can be found as the fractional part of this gene.

The node sequence and sequence of parallel indices can be decoded according to Algorithm 3. The loop in lines

3–14 is executed until the destination node is reached by the path, or any node that only has neighbours that are already in the path. The next node to add to the path is found in line 10, and the parallel index to use is decoded in lines 12–14 and added to the parallel index sequence.

**Algorithm 3:** Decode priority based representations for the multigraph MSPP

```

Input: chromosome,  $G, v_O, v_D$ 
Output: nodes: node sequence, indices: sequence of parallel indices
1  $nodes \leftarrow$  List with a single element:  $v_O$ 
2  $indices \leftarrow$  Empty list
3 while Last element of  $nodes \neq v_D$  do
4    $neighbours \leftarrow$  Set of neighbours in  $G$  of last element in  $nodes$ 
5    $allowed \leftarrow$  Set of elements in  $neighbours$  not in  $nodes$ 
6   if  $allowed$  is empty then
7     Add  $v_D$  to  $nodes$  // Path got stuck, will be penalised
8   else
9      $prevNode \leftarrow$  Last element of  $nodes$ 
10     $nextNode \leftarrow$  Node with maximum priority in  $allowed$ 
11    Add  $nextNode$  to  $nodes$ 
12     $r \leftarrow$  the parallel edge indicator for node  $prevNode$ 
13     $currentIndex \leftarrow \lfloor r * l(prevNode, nextNode) \rfloor + 1$ 
14    Add  $currentIndex$  to  $indices$ 
15 return  $nodes, indices$ 
    
```

An example of a chromosome from this representation that encodes the solution path in Fig. 1 is:

[6.38, 3.67, 1.24, 2.51, 5.09, 4.82].

**Random Key-Based Representation for the Multigraph MSPP**

Given that the priorities in this representation [27] are floating-point numbers, we cannot use the same strategy to include the parallel edge indicators. They are kept separately,

making the genes two-dimensional. The first value of the gene at locus  $i$  encodes the priority value of the node with ID  $i$ , while the second of the gene at locus  $i$  encodes the parallel edge indicator of the node with ID  $i$ . The sequence of nodes and parallel edge indices can be decoded from a chromosome according to Algorithm 3.

An example of a chromosome from this representation that encodes the solution path in Fig. 1 is:

[(0.93, 0.38), (0.36, 0.67), (0.12, 0.24), (0.25, 0.51), (0.51, 0.09), (0.45, 0.82)].

**Variation Operators for the Representations**

Different representations require different genetic operators. This section presents the genetic operators employed in this paper with each of the representations. For some representations, multiple crossover operators are considered, resulting in different variants of the same representations. The variants are summarised in Table 1 with example chromosomes that encode the same solution path for each representation.

There are two variants for the direct variable length representation, the difference being the method used for encoding the parallel edges. Both variants use the same genetic operators. Unlike the simple graph case, crossovers can be conducted on any two parents that have at least one node in common apart from the origin and destination node. Even if the resulting node sequence is the same as a parent node sequence, the parallel edges might change. The mutation operator generates new partial solutions by a random walk from a randomly chosen node in the path.

There is only one variant for the direct fixed length representation. Uniform crossover is employed. In mutation, the integer part of each gene is changed with probability 0.5. The integer part at locus  $i$  is changed to a random neighbour of the  $i$ th node. There is no need to change the fractional part, i.e., the parallel edge indicators, because they indicate a decision between a different set of parallel edges.

**Table 1** Representations and their variants for the multigraph MSPP

Representation	Example candidate	Description	Variants
Direct variable length	[1.02, 6.02, 2.03, 4.02]	Node IDs listed with parallel edge indices	<i>DirVarL-dir</i>
	[1.38, 6.82, 2.67, 4.51]	Node IDs listed with parallel edge indicators	<i>DirVarL-indir</i>
Direct fixed length	[6.38, 4.67, 2.24, 5.51, 4.09, 2.82]	Pointers to neighbouring nodes with parallel edge indicators	<i>DirFixL</i>
Integer priority	[6.38, 3.67, 1.24, 2.51, 5.09, 4.82]	Integer priority values with parallel edge indicators	<i>IntPri-PX, IntPri-WMX</i>
Random keys	[(0.93, 0.38), (0.36, 0.67), (0.12, 0.24), (0.25, 0.51), (0.51, 0.09), (0.45, 0.82)]	Floating point priority values with parallel edge indicators	<i>RanKey-arithX, RanKey-2ptX, RanKey-uniX</i>

Examples are shown for the encoding of the solution path in Fig. 1 using each representation. Variants investigated are listed for each representation. The details of the variants are described in “Variation Operators for the Representations”

We include two variants for the integer priority representation, with two different crossover operators, PX [26] and WMX [10], which was specifically introduced for the MSPP. In both cases, insertion mutation is used, meaning that a randomly picked gene is removed from the chromosome and inserted back at a new locus. We abbreviate the integer-valued priority-based representation for the multigraph MSPP using parallel edge indicators paired with WMX *IntPri-WMX* and *IntPri-PX* when it is paired with PX.

We include three variants for the random key representation paired with three different crossover operators. In Ref. [27], arithmetic crossover was used. We also investigate two-point crossover and uniform crossover. Insertion mutation is used in all these cases. Both the mutation and crossover mechanisms are independent of the values of the genes, and thus, they are straightforward to apply on the two-dimensional genes described in “[Random Key Based Representation for the Multigraph MSPP](#)”. We abbreviate the random key-based representation for the multigraph MSPP using parallel edge indicators *RanKey-arithX*, *RanKey-uniX*, or *RanKey-2ptX* when it is paired with arithmetic, uniform, or two-point crossover, respectively.

### Heuristic Initialisation

#### Heuristic Initialisation Based on Graph Structures

Inspired by [25], we propose a novel heuristic initialisation technique that aims to discourage detours (HeurI1). The method incorporates knowledge about the network structure and randomisation to provide a diverse initial population of high quality. The difference is that we incorporate the roles of IDs and priorities into a single priority value assigned to each node in a semi-random way. The idea is to give higher priorities to nodes closer to the destination node with higher priorities than to the nodes far from it, thereby discouraging detours.

The method is first introduced for random key representation. Then, the initialised solutions can be easily converted to the other three representations. The priority  $p$  is assigned to node  $v$  according to (2). The hopcount (the minimum number of edges in a path) between node  $v$  and node  $v_D$  in  $G$  is denoted  $h(v, v_D, G)$ , and  $\tau_{max}$  denotes the maximum value of the randomisation coefficient  $\tau$ :

$$p(v, v_D) = -h(v, v_D, G) + \tau, \tau \in (0, \tau_{max}). \tag{2}$$

The likelihood of detours appearing in the decoded paths can be controlled by the parameter  $\tau_{max}$ . The higher  $\tau_{max}$  is, the more random the priorities are, and the less prominent is the effect of the heuristic initialisation compared to a purely random one. With a value of  $\tau_{max} > 1$ , small detours are possible, a path might move from a node to another one with the

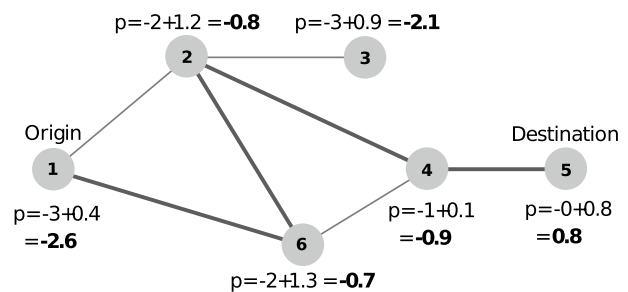
same hopcount, as depicted in Fig. 2. If  $\tau_{max} > 2$ , moving to a node that is at higher hopcount from the destination is possible and becomes more probable with the increase of  $\tau_{max}$ . The value of  $\tau_{max}$  can be optimised with the rest of the parameters for the experiments.

The resulting values need to be transformed according to the representations before they are fed to the algorithms as initial populations. For random key encoding, they need to be normalised to fit the appropriate intervals. For integer-valued priority encoding, the priorities are converted to integers by sorting them into increasing order and assign to each node the rank of its priority value. For direct encodings, the priorities are converted to node-based representations. This way, the method can be used with all four representations. The parallel edge indicators are assigned randomly.

A crucial point is that the heuristic initialisation method should be easily computable compared to the original problem being solved. The proposed method makes use of the hopcount of each node in the graph from the destination node, which can be computed in  $O(V + E)$  time. This is significantly lower than solving the multigraph MSPP, which is in general NP-hard.

#### Heuristic Initialisation Based on Single Objective Search

The second heuristic initialisation method (HeurI2) is adapted from [9], it returns an already good solution by using single objective search (Dijkstra’s algorithm). We use this method to initialise five solutions for the bi-objective problem with weights for the objective values equally distributed in the interval (0, 1). For the triple objective problem, we initialise seven solutions with this method: one with each single objective, one with each pair of objectives with equal weights, and one considering all objectives with equal weights. The rest of the candidates are initialised using the HeurI1 method, or a purely random initialisation.



**Fig. 2** Illustration of the role of  $\tau_{max}$  in heuristic initialisation. Here,  $\tau_{max} = 1.5$ , and thus, detours are possible, as demonstrated by the path indicated in bold, on nodes 1, 6, 2, 4, 5. The path without detour would consist of nodes 1, 6, 4, 5. Because the difference in the hopcounts of nodes 2 and 4 from node 5 is smaller than the difference of the random numbers associated with these nodes, a detour is formed.



The HeurI2 method is most conveniently used with the direct variable length representation, as in Ref. [9], because Dijkstra’s algorithm returns a solution path. Here, we also use it with the other three representations. To do this, we first need to translate the solution path with the parallel indices to the respective representations. There are multiple ways to do this, because only the direct variable length representation provides one-to-one encoding.

In Algorithm 4, we describe one of the possible methods for producing an integer priority-based chromosome that encodes a given solution path. In the first stage, the priorities of the nodes that appear in the path are set (lines 5–10). These priorities are increasing from the destination node towards the origin node. The rest of the genes are filled up with the remaining priority values, and random parallel edge indicators in lines (13–15). This way, at any point in the decoding process, the neighbour that follows the current node in the path has the highest priority in the whole graph except for nodes already in the path. Parallel edge indicators are set to encode the required parallel indices between nodes in the solution path (lines 6–8). The translation to the remaining two representations is done along the same lines.

**Algorithm 4:** Translate sequence of nodes and sequence of parallel edge indices to integer priority based chromosome for multigraph MSPP

```

Input: nodeSequence, indexSequence,  $G, v_D$ 
Output: priorityChromosome: integer priority based representation of same solution
1  $n \leftarrow$  The number of nodes in  $G$ 
2  $priorityChromosome \leftarrow$  Array of length  $n$ , all elements equal  $-1$ 
3  $nextPriority \leftarrow n - 1$ 
4  $length \leftarrow$  length of  $nodeSequence$ 
5 for  $i \leftarrow 1$  to  $length - 1$  do
6      $l \leftarrow$  The number of parallel edges between the  $i$ th and  $(i + 1)$ th nodes in  $G$ .
7      $index \leftarrow$   $i$ th element in  $indexSequence$ 
8      $parallelIndicator \leftarrow \frac{index / l + (index + 1) / l}{2}$ 
9     Replace the element of  $priorityChromosome$  with that corresponds to the  $i$ th node in  $nodeSequence$  with  $(nextPriority + parallelIndicator)$ 
10     $nextPriority \leftarrow nextPriority - 1$ 
11 Replace the element of  $priorityChromosome$  that corresponds to  $v_D$  with  $(nextPriority)$ 
12 for  $i \leftarrow 1$  to  $n$  do
13     if The  $i$ th element of  $priorityChromosome$  is  $-1$  then
14         Replace the  $i$ th element with  $(nextPriority + \text{random number} \in (0, 1))$ 
15          $nextPriority \leftarrow nextPriority - 1$ 
16 return  $priorityChromosome$ 
    
```

### Implementation Details

NSGA-II [29] is employed to compare the representation and initialisation methods. The selection and elitism mechanisms are defined by NSGA-II. This algorithm scales well

**Table 2** Values of the parameters for the different variants, tuned by the irace package

Variants	Population size	Crossover rate	Mutation rate	$\tau_{max}$
DirVarL-dir	64	0.94	0.10	2.46
DirVarL-indir	58	0.65	0.08	2.00
DirFixL	28	0.89	0.04	0.63
IntPri-PX	44	0.62	0.12	0.30
IntPri-WMX	28	0.94	0.15	0.50
RanKey-arithX	64	0.93	0.20	0.65
RanKey-2ptX	38	0.95	0.17	1.02
RanKey-uniX	24	0.99	0.08	2.76

for two and three objective problems [30]. All numerical tests are performed on Queen Mary’s Apocrita HPC facility [31]. The methods are implemented in Python, for the NSGA-II implementation, the inspyred package [32] was used. The parameters for the NSGA-II and the initialisation were tuned with the use of the irace package [33], separately for all the variants of the representations. The tuned parameters are shown in Table 2.

The fitness of a valid path is calculated according to (1). It might happen that some candidates encode paths that do not reach the destination node. In these cases, a penalty function is used, which assigns a large cost to such candidates. The penalty is larger the further away the path ends from the destination node, measured by hopcount. The fitness of an infeasible path  $P'$  that does not reach the destination node is calculated according to Eq. (3), where  $\overline{cost_{max}}$  is the  $k$ -dimensional vector where each component equals the maximum value of any cost component in the given instance:

$$C(P') = \sum_{e \in P'} \overline{cost}(e) + \overline{cost_{max}} * h(P', v_D). \tag{3}$$

### Test Instances

The algorithms are evaluated using 32 test instances, 16 for the 2 objective problem, and 16 for the 3 objective problem. These instances differ in the graph type, the maximum number of parallel edges in the multigraph, and the correlation between the objectives.

We use Waxman networks [34] with 100 and 196 nodes and square grid networks with the same number of nodes (10 by 10 and 14 by 14 nodes). The origin and destination nodes are specified as two endpoints of a diameter (largest hopcount) of the network, to ensure that they are not too close, to avoid setting a trivial problem. Each edge in these

simple graphs is converted to a multi-edge by assigning a cost matrix to it.

The cost matrix contains a cost-vector in each of its rows, corresponding to a parallel edge. The number of rows of these cost matrices is randomly chosen between 1 and  $l_{\max}$ , where  $l_{\max}$  the maximum allowed number of parallel edges, 5 or 10 in this case. All parallel edges between the same two nodes have non-dominated cost-vectors.

We included instances with uncorrelated objectives and with negative correlation between the objectives. Negative correlation is the case where a multi-objective approach is essential for real-world applications and also these kinds of instances are the most challenging for exact solution approaches [35].

The cost assignment method described in Ref. [35] was used, to generate pairs of cost components with a specified negative correlation multiplier  $\rho$ . The method is described by (4), where the first cost component ( $C_1$ ) is randomly generated from a uniform distribution within the interval specified by  $C_{\min}$  and  $C_{\max}$ .  $(C_2)^*$  is a randomly generated value from the same interval, and  $C_2$  is the second cost component. In our instances, the interval is specified by  $C_{\min} = 10$  and  $C_{\max} = 1000$ :

$$C_2 = C_{\max} + C_{\min} - (|\rho| * C_1 + (1 - |\rho|) * (C_2)^*). \quad (4)$$

Note that for positive correlation multipliers, there is a separate formula [36]. However, we do not include instances with positive correlation, because they are generally easier to solve by exact approaches. In the case of three objectives, the third cost component is also calculated from the first cost component with the same value of  $\rho$ .

## Evaluation of Approximate Solutions

The proposed representations and their variants (listed in Table 1) are tested empirically and their performances are compared to a reference front using a set of quality indicators. The reference front is the true Pareto front, when it is available, and an approximation of it otherwise.

### Reference Fronts

The true Pareto front was found by a state-of-the-art exact algorithm NAMOA\* [37], a multi-objective variant of the A\* algorithm, which was adapted to the multigraph problem. This algorithm uses heuristic functions to speed up the search. Here, we used a heuristic function proposed in [38] defined as,  $h_{TC}(n) = (c_1(n), c_2(n), \dots, c_q(n))$ , where  $c_i(n)$  is the optimal scalar cost of a path from node  $n$  to the destination node, considering only the  $i$ th cost component.

We allowed 1 day for the execution time of NAMOA\*. When this was not enough, we approximated the Pareto

front using the solutions already found by NAMOA\*, if any, and the approximate solutions returned by variants of NSGA-II. There are some bounds on the quality of Pareto fronts approximated this way. We know that they contain at least as many members of the true Pareto front as the number of objectives, because solutions initialised with the HeurI2 method are included. When NAMOA\* is stopped prematurely, it either does not return any solutions, or it returns a subset of the Pareto-optimal solutions, these are also included in the reference front when they are available. Thus, all other members of the approximate front are non-dominated by at least some members of the true Pareto front.

### Quality Indicators

We use the multiplicative Epsilon indicator, the R3 indicator, and the relative hypervolume (RHV) indicator to evaluate the approximations of the Pareto fronts. A detailed description of these indicators can be found in Ref. [39]. The use of these metrics is in line with the recommendations in [40]. All three indicators signal higher qualities of the approximation front by lower values. When the approximate front is fully converged to the real Pareto front, the R3 metric and the RHV indicators have a value of 0, and the Epsilon indicator has a value of 1.

## Results

### Finding Exact Solutions for the Multigraph MSPP Instances

Table 3 presents the experimental data regarding solving the 32 test instances using NAMOA\*. In particular, it shows the high execution times of NAMOA\*; for grid graphs, this time often exceeds 24 h, and overall, it is only twice below 10 s.

We can see the size of the reference front (true Pareto front or the approximated Pareto front). According to both running time and reference front size, the bi-objective instances on Waxman graphs can be considered the easiest and the instances on grid graphs with three objectives the most difficult. The last two columns also show that if NAMOA\* is stopped prematurely after 10 or 20 s, it does not return any solutions in the case of most test instances. When it does return solutions after this short time, we can see that they are rarely a good approximation of the reference front, as measured by the RHV quality metric. These results are compared to NSGA-II in “[Differences Across the Instances](#)”.

In the following, we present results about using NSGA-II with the different representations and show that high-quality approximate solutions can be found in only 10 s.

**Initialisation Methods**

Four different initialisation procedures were used.

- *R*: Purely random initialisation
- *H1*: HeurI1 for all candidates

- *H2 + R*: HeurI2 for 5 or 7 candidates as described in “Heuristic Initialisation”, and purely random initialisation for the rest
- *H1 + H2*: HeurI2 for 5 or 7 candidates and HeurI1 for the rest.

We combined all eight variants of NSGA-II with each initialisation procedure and compared the achieved solution qualities according to the quality indicators in Table 4. We can see that H1 + H2 and H2 + R are visibly better than the

**Table 3** The performance of NAMOA\* on the 32 problem instances. The first five columns describe the problem instances

Problem instances					NAMOA*	Reference	NAMOA*	NAMOA*
Obj	Graph type	$l_{max}$	n	$\rho$	Time (s)	Front size	RHV 10 s	RHV 20 s
2	Grid	5	100	- 0.75	8601.61	1247	-	-
				0.00	726.40	236	-	-
			- 0.75	-	592	-	-	
		10	100	0.00	14,917.48	472	-	-
				- 0.75	62,779.48	2599	-	-
			0.00	3536.90	416	-	-	
	Waxman	5	100	- 0.75	-	506	-	-
				0.00	14,387.35	657	-	-
			0.00	6.68	36	0.000	0.000	
		10	100	- 0.75	11.59	48	0.072	0.000
				0.00	305.88	236	0.412	0.292
			- 0.75	61.50	57	0.231	0.173	
3	Grid	5	100	- 0.75	442.22	155	0.258	0.233
				0.00	9.97	69	0.000	0.000
			- 0.75	1787.01	368	-	-	
		10	100	0.00	256.54	76	-	-
				- 0.75	-	76	-	-
			0.00	-	1961	-	-	
	Waxman	5	100	- 0.75	-	1316	-	-
				0.00	-	2536	-	-
			- 0.75	-	1006	-	-	
		10	100	- 0.75	-	1806	-	-
				0.00	-	1355	-	-
			- 0.75	-	3518	-	-	
Waxman	5	100	0.00	-	1357	-	-	
			- 0.75	839.94	218	-	-	
		0.00	205.06	190	-	0.405		
	10	100	- 0.75	1654.18	567	0.609	0.590	
			0.00	3043.82	258	0.290	0.290	
		- 0.75	3097.27	868	0.611	0.583		
10	100	0.00	7178.79	651	0.445	0.390		
		- 0.75	-	946	-	-		
				0.00	70,164.86	666	-	-

The column “NAMOA\* time” specifies the running time of the NAMOA\* algorithm on the given instances in seconds, if this time is less than a day. The column “Reference front size” specifies the number of solutions in the Pareto front, or when the algorithm does not finish in a day, the number of solutions in the approximation of the Pareto front. The last two columns describe the solution quality measured by relative hypervolume reached by NAMOA\* when it is stopped prematurely after 10 or 20 s

other two in all cases. One main reason for this is that the H2 method finds the extreme solutions in the initialisation step, while these are often not found at all using a method without H2.

Apart from the direct variable length representation, the H1 + H2 method significantly outperforms the H2 + R. On the other hand, H2 + R is significantly better than H1 + H2 only in case of the DirVarL-dir variant. The  $p$  values for the statistical tests are indicated in Table 4. The general patterns of the results are similar in case of the 20 s time budget. For details, see the Appendix Table 6.

One possible explanation for why the H1 + H2 initialisation method is not better than the H2 + R method only in case of the direct variable length representation is the following. In the case of the other three representations, the H1 method encodes some extra information about which direction to choose at each node in the graph, not just the nodes in the encoded path. This is possible because of the inherent ambiguity of these representations. If such a chromosome is modified through mutation or crossover, the extra information might make it more probable that the resulting new chromosome will be a valid path. While the direct variable length representation encodes the path without ambiguity, and thus, the extra information provided by the H1 initialisation is lost when the chromosome is converted to this representation.

## Comparison of NSGA-II Variants

In the following, only the best initialisations are considered for each variant.

### Performance of Variants on Average Across All Instances

Table 4 also shows that, on average, the RanKey-2ptX variant reaches the best results (with H1 + H2 initialisation) considering all instances. This is confirmed by the one-sided Wilcoxon signed-rank test with  $p$  values below  $10^{-7}$  for all three quality indicators. This also holds for the 20 s time budget, see Appendix Tables 6 and 7.

These results might be surprising given that RanKey-2ptX has the most ambiguity out of the four representations. The results suggest that a different search landscape and more effective genetic operators can balance the increased ambiguity of representations. The extra information encoded in the chromosomes with some ambiguity—as discussed in “Initialisation Methods”—might also contribute.

In the following, we identify the best variant for each of the four representations, based on the data presented in Table 4 for the further analysis.

There were two variants considered for the direct variable length representation. The DirVarL-dir variant was better according to the averages of all three quality indicators than the DirVarL-indir. The one-sided Wilcoxon signed-rank test

**Table 4** Illustration of the role of the heuristic initialisation methods in the solution quality achieved by variants of NSGA-II using different genetic representations and crossover operators, described by the R3 metric

Representation	DirFixL	DirVarL	DirVarL	IntPri	IntPri	RanKey	RanKey	RanKey
Initialisation		-dir	-indir	-PX	-WMX	-2ptX	-arithX	-uniX
Epsilon indicator								
R	7.7077	3.3283	3.6321	3.7194	2.7993	3.5390	4.0898	4.1144
H1	2.7818	3.1189	3.3199	2.9096	2.2668	2.2565	2.7872	2.3880
H1 + H2	<b>1.2041***</b>	1.2144	1.2358	<b>1.3228***</b>	<b>1.2897***</b>	<b>1.1915***</b>	<b>1.2458***</b>	<b>1.2547***</b>
H2 + R	1.2487	<b>1.2090*</b>	<b>1.2334</b>	1.3368	1.2998	1.2039	1.2731	1.2771
R3 indicator								
R	0.6012	0.3682	0.3975	0.3961	0.3122	0.3860	0.4631	0.4344
H1	0.2913	0.3428	0.3625	0.3173	0.2389	0.2244	0.3136	0.2481
H1 + H2	<b>0.0255***</b>	0.0257	0.0285	<b>0.0439***</b>	<b>0.0394***</b>	<b>0.0231***</b>	<b>0.0359***</b>	<b>0.0346***</b>
H2 + R	0.0300	<b>0.0246**</b>	<b>0.0283</b>	0.0466	0.0410	0.0241	0.0396	0.0375
RHH indicator								
R	0.5582	0.2296	0.2571	0.2652	0.1799	0.2439	0.3162	0.2971
H1	0.1632	0.2071	0.2231	0.1864	0.1230	0.1121	0.1690	0.1279
H1 + H2	<b>0.0135***</b>	0.0140	<b>0.0160</b>	<b>0.0269***</b>	<b>0.0227***</b>	<b>0.0124***</b>	<b>0.0192***</b>	<b>0.0196***</b>
H2 + R	0.0180	<b>0.0137*</b>	<b>0.0160</b>	0.0302	0.0241	0.0132	0.0223	0.0229

The lowest values of the quality indicators corresponding to the best initialisation technique are highlighted in bold in each sub-column

Lower values correspond to better quality. The average values are calculated from 50 runs for each of the 32 instances. The time budget for NSGA-II is specified as 10 s. The one-sided Wilcoxon signed-rank test was used to decide statistical significance between H1 + H2 and H2 + R. Results are indicated as (\*) $p < 0.05$ , (\*\*)  $p < 0.005$ , and (\*\*\*)  $p < 0.0005$

confirms the statistical significance in all cases with  $p$  values below  $10^{-35}$  for both values of the time budget. This difference confirms the expectations that increasing the ambiguity of the representation of the parallel edge indices decreases the effectiveness of the search to some degree.

There were two variants considered for the integer priority representation. The IntPri-WMX variant outperformed the IntPri-PX variant according to all quality indicators on average. The one-sided Wilcoxon signed-rank test confirms the statistical significance in all cases with  $p$  values below  $10^{-22}$  for both values of the time budget. This is not surprising given that WMX was specifically introduced for the shortest path problem, while PX not.

There were three variants considered for the random key representation. The RanKey-2ptX variant outperformed both the RanKey-arithX and RanKey-uniX variants among all others, as we have already seen in the overall comparison of the variants.

Taking the average of the quality indicators across all instances as the basis of comparison might cover up important differences between the representations. It can be the case that the RanKey-2ptX variant is consistently outperformed by another variant for a subgroup of the instances. This is investigated next.

### Performance Differences Across the Instances

In the following, we only consider the best variant for each representation with the best initialisation method. The four representations are compared in detail on the 32 instances separately. Here, the representations are compared with a time budget of 10 s. For the sake of comparison, the solution quality achieved by NAMOA\* in 10 s is also included. The RHV values are shown, because this quality indicator has the highest correlation with the other two. Note that the value of the Spearman's rank correlation coefficient is above 0.91 between any two of the three quality indicators, and this suggests that using any of the three would result in roughly the same order between the evaluated variants.

In Table 5, we can see that while RanKey was shown to be the best on average in “[Performance of Variants on Average Across All Instances](#)”, in fact, three of the representations are shown to be competitive when compared separately for each of the instances. IntPri-WMX is outperformed by the other three in the majority of cases. DirVarL-dir seems to be particularly suitable for grid instances with three objectives, while generally less successful in case of Waxman networks and grids with two objectives. RanKey-2ptX shows an overall good performance. However, the grid instances with more than two objectives seem to be a weak point.

We can also see in Table 5 that NAMOA\* outperformed NSGA-II with the 10 s time budget in only two cases, when it found the whole Pareto front within 10 s. In other cases,

even if NAMOA\* returns some solutions, the quality is magnitudes lower than with the NSGA-II variants. Thus, we can conclude that for these test instances, we should not choose NAMOA\* as a solver algorithm if the time budget is small.

The general patterns of the results are similar with the other quality indicators and with the time budget of 20 s. For the detailed results, see Appendix Tables 6, 7, 8, 9, 10, 11, and 12.

## Conclusion and Future Work

A systematic investigation of design choices of genetic representations for the multigraph MSPP was presented in this paper. Four main representations were investigated, some with multiple variants. The resulting eight variants were compared using NSGA-II.

The proposed encoding schemes for incorporating the choice between parallel edges are through a floating-point number as parallel edge indicators, instead of including the parallel edge indices directly. This approach is necessary for the extension of priority-based representations to the multigraph MSPP when inhomogeneous numbers of parallel edges between pairs of nodes are possible.

Multiple different initialisation methods are also considered. Apart from purely random initialisation, two heuristic initialisation methods are investigated. One of these is an existing method based on single objective search that was previously only used with the direct variable length representation. We adapted it to be applicable to the other representations. We also propose a novel heuristic initialisation method that makes use of a priori knowledge about the graph structure and can be used with all representations.

The representations and their variants combined with different initialisation methods are compared according to three quality indicators. It is found that our proposed initialisation method provides additional benefits when used together with the other heuristic initialisation methods in three out of the four representations.

Regarding the average of the quality indicators across all instances, the best variant of the algorithm was the random keys representation with two-point crossover. This variant employs the novel heuristic initialisation proposed, and the parallel edge indicators to encode the choice between parallel edges, which proves the effectiveness of the proposed methods.

When breaking down the test instances further, it was revealed that for grid-based problem instances with more than two objectives, the best approach is the direct variable length representation, while the random keys representation is generally the best for the other problem instances. This might be because the crossover operator of the direct variable length representation requires a common node between



**Table 5** Comparing the best variants of NSGA-II for the four representations, separately for the 32 test instances with a time budget of 10 s

Problem instance					DirVarL-dir	DirFixL	RanKey-2ptX	IntPri-WMX	NAMOA* 10 s
Obj	Graph type	$l_{\max}$	$n$	$\rho$					
2	Grid	5	100	-0.75	0.0100***	<b>0.0070</b>	<b>0.0070</b>	0.0122***	-
				0.00	0.0109***	0.0101***	<b>0.0062</b>	0.0152***	-
		10	100	-0.75	0.0125***	0.0078	<b>0.0077</b>	0.0134***	-
				0.00	0.0103***	0.0146***	<b>0.0079</b>	0.0217***	-
		196	100	-0.75	0.0076***	<b>0.0064</b>	<b>0.0064</b>	0.0128***	-
				0.00	<b>0.0058</b>	0.0086***	0.0067*	0.0169***	-
	Waxman	5	100	-0.75	0.0017***	<b>0.0007</b>	0.0012***	0.0009*	0.0000
				0.00	0.0072***	0.0057***	<b>0.0025</b>	0.0027	0.0718
		10	100	-0.75	0.0107***	0.0031	<b>0.0030</b>	0.0089***	0.4118
				0.00	0.0052***	<b>0.0019</b>	0.0034***	0.0107***	0.2314
		196	100	-0.75	0.0026***	0.0025**	<b>0.0022</b>	0.0026***	0.2579
				0.00	0.0089***	0.0046***	<b>0.0037</b>	0.0104***	0.0000
3	Grid	5	100	-0.75	<b>0.0082</b>	0.0150***	0.0140***	0.0209***	-
				0.00	<b>0.0222</b>	0.0259***	0.0261***	0.0584***	-
		10	100	-0.75	<b>0.0122</b>	0.0183***	0.0176***	0.0267***	-
				0.00	<b>0.0229</b>	0.0485***	0.0387***	0.0688***	-
		196	100	-0.75	<b>0.0129</b>	0.0136	0.0150***	0.0215***	-
				0.00	<b>0.0178</b>	0.0374***	0.0397***	0.0659***	-
	Waxman	5	100	-0.75	<b>0.0072</b>	0.0200***	0.0193***	0.0272***	-
				0.00	<b>0.0176</b>	0.0303***	0.0324***	0.0744***	-
		10	100	-0.75	0.0189***	0.0122***	<b>0.0091</b>	0.0106***	-
				0.00	0.0104***	<b>0.0075</b>	0.0079	0.0156***	-
		196	100	-0.75	0.0161***	0.0078***	<b>0.0055</b>	0.0128***	0.6088
				0.00	0.0366***	0.0153**	<b>0.0139</b>	0.0255***	0.2895
196	100	-0.75	<b>0.0061</b>	0.0103***	0.0082***	0.0117***	0.6114		
		0.00	<b>0.0169</b>	<b>0.0169</b>	0.0193**	0.0285***	0.4449		
			-0.75	0.0421***	0.0174**	<b>0.0151</b>	0.0296***	-	
			0.00	0.0426***	0.0283	<b>0.0276</b>	0.0488***	-	

The best values among the representations are shown in bold in each row

The values of the average RHV quality indicator of 50 runs with each representation is shown. The results of NAMOA\* regarding the RHV quality indicator from Table 3 with the same time budget of 10 s are repeated in the last column, for easy comparison. The one-sided Wilcoxon signed-rank test was used to decide statistical significance between each variant compared to the best one in each row. Results are indicated as (\*) $p < 0.05$ , (\*\*)  $p < 0.005$ , (\*\*\*)  $p < 0.0005$

the two parent paths, and this is more likely to exist in grid graphs. One future research direction is to understand this difference in depth, and design an algorithm that incorporates the strengths of both representations.

The solution qualities achieved this way gave a good representation of the reference fronts in only 10 s, while obtaining the exact Pareto front required significantly longer times for most of the investigated instances. The running times are often important in practical applications.

We saw that the ambiguity of representing the choice of parallel edges makes NSGA-II less effective. However, the inherent ambiguity of the random key representation in encoding node sequences does not seem to cause a problem. One future direction is to understand how can the representation with the most amount of ambiguity be the most successful.

Additionally, future work includes the extension to constrained problems, particularly the investigation of constraint handling methods. The extension will allow

the proposed approaches to be tested on real-world problems and to be compared with other exact and approximate algorithms.

## Appendix

See Tables 6, 7, 8, 9, 10, 11, and 12.

**Table 6** Illustration of the role of the heuristic initialisation methods in the solution quality achieved by variants of NSGA-II using different genetic representations and crossover operators, described by the R3 metric

Representation	DirFixL	DirVarL	DirVarL	IntPri	IntPri	RanKey	RanKey	RanKey
Initialisation		-dir	-indir	-PX	-WMX	-2ptX	-arithX	-uniX
<i>Epsilon indicator</i>								
R	7.4730	3.1227	3.4750	3.2724	2.5375	3.4757	3.9342	3.9197
H1	2.7997	2.9739	3.1787	2.6662	2.1416	2.1760	2.7162	2.3856
H1 + H2	<b>1.1934***</b>	1.2048	1.2263	<b>1.2909***</b>	<b>1.2738***</b>	<b>1.1765***</b>	<b>1.2371***</b>	<b>1.2358***</b>
H2 + R	1.2363	<b>1.2000***</b>	<b>1.2225</b>	1.3045	1.2787	1.1853	1.2621	1.2511
<i>R3 indicator</i>								
R	0.5846	0.3485	0.3825	0.3548	0.2778	0.3822	0.4504	0.4230
H1	0.2904	0.3323	0.3495	0.2861	0.2192	0.2171	0.3077	0.2481
H1 + H2	<b>0.0236***</b>	0.0239	0.0268	<b>0.0389**</b>	<b>0.0364***</b>	<b>0.0202***</b>	<b>0.0342***</b>	<b>0.0315***</b>
H2 + R	0.0281	<b>0.0231***</b>	<b>0.0262</b>	0.0406	0.0376	0.0212	0.0380	0.0340
<i>RHV indicator</i>								
R	0.5353	0.2111	0.2415	0.2238	0.1525	0.2418	0.2998	0.2840
H1	0.1633	0.1957	0.2111	0.1632	0.1095	0.1065	0.1634	0.1264
H1 + H2	<b>0.0124***</b>	0.0129	<b>0.0148</b>	<b>0.0232***</b>	<b>0.0208***</b>	<b>0.0107***</b>	<b>0.0181***</b>	<b>0.0174***</b>
H2 + R	0.0163	<b>0.0124***</b>	<b>0.0148</b>	0.0254	0.0217	0.0113	0.0210	0.0201

The lowest values of the quality indicators corresponding to the best initialisation technique are highlighted in bold in each sub-column

Lower values correspond to better quality. The average values are calculated from 50 runs for each of the 32 instances. The time budget for NSGA-II is specified as 20 s. The one-sided Wilcoxon signed-rank test was used to decide statistical significance between H1 + H2 and H2 + R. Results are indicated as (\*) $p < 0.05$ , (\*\*)  $p < 0.005$ , (\*\*\*)  $p < 0.0005$

**Table 7** Comparing variants of NSGA-II with their best initialisation methods for both the 10 s and 20 s time budgets

Representation	DirFixL	DirVarL	DirVarL	IntPri	IntPri	RanKey	RanKey	RanKey
Time budget [s]		-dir	-indir	-PX	-WMX	-2ptX	-arithX	-uniX
<i>Epsilon indicator</i>								
10	1.2041***	1.2090***	1.2334***	1.3228***	1.2897***	<b>1.1915</b>	1.2458***	1.2547***
20	1.1934***	1.2000***	1.2225***	1.2909***	1.2738***	<b>1.1765</b>	1.2371***	1.2358***
<i>R3 indicator</i>								
10	0.0255***	0.0246***	0.0283***	0.0439***	0.0394***	<b>0.0231</b>	0.0359***	0.0346***
20	0.0236***	0.0231***	0.0262***	0.0389***	0.0364***	<b>0.0202</b>	0.0342***	0.0315***
<i>RHV indicator</i>								
10	0.0135***	0.0137***	0.0160***	0.0269***	0.0227***	<b>0.0124</b>	0.0192***	0.0196***
20	0.0124***	0.0124***	0.0148***	0.0232***	0.0208***	<b>0.0107</b>	0.0181***	0.0174***

The one-sided Wilcoxon signed rank test was used to decide statistical significance between each variant compared to the best one in each row (highlighted in bold)

Average solution qualities on the 32 instances are shown according to the three quality indicators. Lower values correspond to better quality. The average values are calculated from 50 runs for each of the 32 instances. The one-sided Wilcoxon signed-rank test was used to decide statistical significance between each variant compared to the best one in each row. Results are indicated as (\*) $p < 0.05$ , (\*\*)  $p < 0.005$ , and (\*\*\*)  $p < 0.0005$

**Table 8** Comparing the best variants of NSGA-II for the four representations, separately for the 32 test instances with a time budget of 20 s

Problem instance					DirVarL-dir	DirFixL	RanKey-2ptX	IntPri-WMX	NAMOA* 20s	
obj	graph type	$l_{\max}$	n	$\rho$						
2	grid	5	100	- 0.75	0.0093***	0.0070***	<b>0.0053</b>	0.0109***	–	
				0.00	0.0101***	0.0092***	<b>0.0050</b>	0.0133***	–	
		10	100	- 0.75	0.0102***	0.0064*	<b>0.0059</b>	0.0120***	–	
				0.00	0.0097***	0.0123***	<b>0.0062</b>	0.0202***	–	
		196	100	- 0.75	0.0068***	0.0056	<b>0.0054</b>	0.0112***	–	
				0.00	0.0056*	0.0077***	<b>0.0050</b>	0.0159***	–	
	Waxman	5	100	- 0.75	0.0060***	0.0055***	<b>0.0046</b>	0.0101***	–	
				0.00	<b>0.0082</b>	0.0137***	0.0114***	0.0185***	–	
		10	100	- 0.75	0.0013***	<b>0.0008</b>	0.0011	<b>0.0008*</b>	0.0000	
				0.00	0.0062***	0.0046***	<b>0.0023</b>	<b>0.0023</b>	0.0000	
		196	100	- 0.75	0.0083***	0.0033***	<b>0.0024</b>	0.0074***	0.2918	
				0.00	0.0045***	<b>0.0018</b>	0.0030***	0.0098***	0.1730	
196	100	- 0.75	0.0024**	0.0026**	<b>0.0022</b>	0.0023*	0.2328			
		0.00	0.0094***	0.0045***	<b>0.0034</b>	0.0100***	0.0000			
3	grid	5	100	- 0.75	<b>0.0069</b>	0.0144***	0.0124***	0.0194***	–	
				0.00	<b>0.0204</b>	0.0239***	0.0229***	0.0543***	–	
		10	100	- 0.75	<b>0.0108</b>	0.0165***	0.0156***	0.0234***	–	
				0.00	<b>0.0205</b>	0.0425***	0.0316***	0.0637***	–	
		196	100	- 0.75	0.0125	0.0128	<b>0.0124</b>	0.0192***	–	
				0.00	<b>0.0157</b>	0.0348***	0.0351***	0.0601***	–	
		196	100	- 0.75	<b>0.0062</b>	0.0176***	0.0173***	0.0252***	–	
				0.00	<b>0.0165</b>	0.0259***	0.0274***	0.0707***	–	
		Waxman	5	100	- 0.75	0.0162***	0.0121***	<b>0.0084</b>	0.0098***	–
					0.00	0.0094***	0.0069	<b>0.0067</b>	0.0143***	0.4047
			10	100	- 0.75	0.0122***	0.0077***	<b>0.0049</b>	0.0118***	0.5902
					0.00	0.0344***	0.0158***	<b>0.0120</b>	0.0219***	0.2895
	196		100	- 0.75	<b>0.0052</b>	0.0105***	0.0077***	0.0106***	0.5829	
				0.00	0.0165**	<b>0.0151</b>	0.0190***	0.0258***	0.3898	
	196	100	- 0.75	0.0376***	0.0176***	<b>0.0127</b>	0.0259***	–		
			0.00	0.0403***	0.0269	<b>0.0238</b>	0.0462***	–		

The best values among the representations are shown in bold in each row

The values of the average RHV quality indicator of 50 runs with each representation are shown. The results of NAMOA\* regarding the RHV quality indicator with the same time budget of 20 s are shown in the last column, for comparison. The one-sided Wilcoxon signed-rank test was used to decide statistical significance between each variant compared to the best one in each row. Results are indicated as (\*) $p < 0.05$ , (\*\*)  $p < 0.005$ , and (\*\*\*)  $p < 0.0005$

**Table 9** Comparing the best variants of NSGA-II for the four representations, separately for the 32 test instances with a time budget of 10 s

Problem instance					DirVarL-dir	DirFixL	RanKey-2ptX	IntPri-WMX	NAMOA* 10 s
Obj	Graph type	$l_{max}$	$n$	$\rho$					
2	Grid	5	100	- 0.75	1.0654***	<b>1.0450</b>	1.0452	1.0638***	-
				0.00	1.1896***	1.2170***	<b>1.1509</b>	1.3325***	-
		10	100	- 0.75	1.1080***	1.0571*	<b>1.0527</b>	1.0735***	-
				0.00	1.2094***	1.3056***	<b>1.1666</b>	1.3539***	-
		196	100	- 0.75	1.0680***	1.0551	<b>1.0532</b>	1.0849***	-
				0.00	<b>1.1630</b>	1.2795***	1.2115***	1.3453***	-
	Waxman	5	100	- 0.75	1.0634***	<b>1.0320</b>	1.0466***	1.0339	1.0000
				0.00	1.1912***	1.1465***	1.0939	<b>1.0934</b>	2.3609
		10	100	- 0.75	1.1301***	<b>1.0536</b>	1.0595**	1.0987***	4.7292
				0.00	1.1884***	<b>1.1003</b>	1.1601***	1.2524***	5.1258
		196	100	- 0.75	1.1088***	<b>1.1058</b>	1.1075**	1.1071***	4.2204
				0.00	1.4441***	<b>1.3008</b>	1.3205	1.5825***	1.0000
3	Grid	5	100	- 0.75	<b>1.0476</b>	1.0596***	1.0599***	1.0778***	-
				0.00	<b>1.2156</b>	1.2329***	1.2189	1.3434***	-
		10	100	- 0.75	<b>1.0781</b>	1.0810*	1.0798*	1.1374***	-
				0.00	<b>1.2009</b>	1.3451***	1.2689***	1.3512***	-
		196	100	- 0.75	<b>1.1489</b>	1.1598	1.2072***	1.2144***	-
				0.00	<b>1.2249</b>	1.3985***	1.3746***	1.6090***	-
	Waxman	5	100	- 0.75	<b>1.0541</b>	1.0989***	1.1010***	1.1653***	-
				0.00	<b>1.2516</b>	1.4188***	1.3974***	1.6710***	-
		10	100	- 0.75	1.2318***	<b>1.2007</b>	1.2124***	1.2142***	-
				0.00	1.2017***	<b>1.1554</b>	1.1567	1.2294***	-
		196	100	- 0.75	1.1151***	1.0546***	<b>1.0458</b>	1.0704***	3.1893
				0.00	1.4183***	1.2640	<b>1.2527</b>	1.3235***	4.7317
196	100	- 0.75	1.0675	1.0700	<b>1.0673</b>	1.0797***	3.7121		
		0.00	<b>1.5491</b>	1.5555	1.6199*	1.7931***	15.2834		
196	100	- 0.75	1.3641***	1.1619*	<b>1.1292</b>	1.2875***	-		
		0.00	1.8879***	<b>1.5401</b>	1.5517	2.1514***	-		

The best values among the representations are shown in bold in each row

The values of the average Epsilon quality indicator of 50 runs with each representation is shown. The results of NAMOA\* regarding the Epsilon quality indicator with the same time budget of 10 s are shown in the last column, for comparison. The one-sided Wilcoxon signed-rank test was used to decide statistical significance between each variant compared to the best one in each row. Results are indicated as (\*) $p < 0.05$ , (\*\*)  $p < 0.005$ , and (\*\*\*)  $p < 0.0005$

**Table 10** Comparing the best variants of NSGA-II for the four representations, separately for the 32 test instances with a time budget of 20 s

Problem Instance					DirVarL-dir	DirFixL	RanKey-2ptX	IntPri-WMX	NAMOA* 20s
Obj	Graph type	$l_{\max}$	n	$\rho$					
2	Grid	5	100	- 0.75	1.0629***	1.0460***	<b>1.0372</b>	1.0585***	–
				0.00	1.1763***	1.2075***	<b>1.1276</b>	1.3225***	–
				- 0.75	1.0827***	1.0512***	<b>1.0431</b>	1.0660***	–
		10	100	0.00	1.2008***	1.2637***	<b>1.1276</b>	1.3542***	–
				- 0.75	1.0631***	1.0503***	<b>1.0446</b>	1.0765***	–
				0.00	1.1610	1.2572***	<b>1.1558</b>	1.3360***	–
	Waxman	5	100	- 0.75	1.0456	1.0464	<b>1.0444</b>	1.0643***	–
				0.00	<b>1.3280</b>	1.7837	1.6194	1.7287***	–
				- 0.75	1.0540***	1.0348	1.0440***	<b>1.0316</b>	1.0000
		10	100	0.00	1.1723***	1.1248***	1.0847	<b>1.0843</b>	1.0000
				- 0.75	1.0988***	<b>1.0536</b>	1.0549	1.0878***	3.8131
				0.00	1.1739***	<b>1.0944</b>	1.1550***	1.2450***	4.1623
3	Grid	5	100	- 0.75	1.1087***	<b>1.1035</b>	1.1074*	1.1069***	3.9561
				0.00	1.4674***	<b>1.2797</b>	1.2969	1.5488***	1.0000
				- 0.75	1.0860***	<b>1.0520</b>	1.0555	1.0611***	–
		10	100	0.00	1.2275***	1.1622***	<b>1.1344</b>	1.2238***	–
				- 0.75	<b>1.0434</b>	1.0562***	1.0556***	1.0750***	–
				0.00	1.2128**	1.2135*	<b>1.2009</b>	1.3173***	–
Waxman	5	100	- 0.75	1.0731	1.0714*	<b>1.0683</b>	1.1146***	–	
			0.00	<b>1.1776</b>	1.3229***	1.2379***	1.3307***	–	
			- 0.75	1.1483*	<b>1.1471</b>	1.1608*	1.1888***	–	
	10	100	0.00	<b>1.2214</b>	1.3661***	1.3490***	1.5684***	–	
			- 0.75	<b>1.0466</b>	1.0926***	1.0846***	1.1476***	–	
			0.00	<b>1.2390</b>	1.3881***	1.3575***	1.6481***	–	
Waxman	5	100	- 0.75	1.2193***	<b>1.1998</b>	1.2012	1.2095**	–	
			0.00	1.1976***	<b>1.1468</b>	1.1512*	1.2200***	5.4526	
			- 0.75	1.0910***	1.0543***	<b>1.0450</b>	1.0670***	3.1091	
	10	100	0.00	1.3924***	1.2732***	<b>1.2353</b>	1.3196***	4.7317	
			- 0.75	<b>1.0640</b>	1.0734**	1.0649	1.0748***	3.6113	
			0.00	1.5479**	<b>1.4930</b>	1.6691***	1.7472***	13.0856	
196	100	- 0.75	1.3506***	1.1594**	<b>1.1117</b>	1.2578***	–		
		0.00	1.8648***	<b>1.5212</b>	1.5232	2.0806***	–		

The best values among the representations are shown in bold in each row

The values of the average Epsilon quality indicator of 50 runs with each representation is shown. The results of NAMOA\* regarding the Epsilon quality indicator with the same time budget of 20 s are shown in the last column, for comparison. The one-sided Wilcoxon signed-rank test was used to decide statistical significance between each variant compared to the best one in each row. Results are indicated as (\*) $p < 0.05$ , (\*\*)  $p < 0.005$ , and (\*\*\*)  $p < 0.0005$



**Table 11** Comparing the best variants of NSGA-II for the four representations, separately for the 32 test instances with a time budget of 10 s

Problem instance					DirVarL-dir	DirFixL	RanKey-2ptX	IntPri-WMX	NAMOA* 10 s
Obj	Graph type	$l_{max}$	n	$\rho$					
2	Grid	5	100	- 0.75	0.0123***	<b>0.0091</b>	<b>0.0091</b>	0.0173***	-
				0.00	0.0283***	0.0253***	<b>0.0170</b>	0.0327***	-
		10	100	- 0.75	0.0156***	0.0108*	<b>0.0100</b>	0.0177***	-
				0.00	0.0275***	0.0341***	<b>0.0226</b>	0.0485***	-
		196	100	- 0.75	0.0170***	<b>0.0149</b>	0.0152	0.0297***	-
				0.00	<b>0.0285</b>	0.0371***	0.0301	0.0596***	-
	Waxman	5	100	- 0.75	0.0016***	<b>0.0007</b>	0.0012***	0.0010***	0.0000
				0.00	0.0094***	0.0076***	<b>0.0035</b>	0.0036	0.0633
		10	100	- 0.75	0.0169***	<b>0.0054</b>	<b>0.0054</b>	0.0137***	0.2933
				0.00	0.0141***	<b>0.0056</b>	0.0094***	0.0299***	0.1820
		196	100	- 0.75	0.0074***	0.0074**	<b>0.0066</b>	0.0078***	0.2288
				0.00	0.0294***	0.0181***	<b>0.0103</b>	0.0320***	0.0000
3	Grid	5	100	0.75	<b>0.0094</b>	0.0161***	0.0147***	0.0223***	-
				0.00	<b>0.0280</b>	0.0341***	0.0330***	0.0639***	-
		10	100	- 0.75	<b>0.0120</b>	0.0194***	0.0190***	0.0296***	-
				0.00	<b>0.0303</b>	0.0637***	0.0520***	0.0831***	-
		196	100	- 0.75	<b>0.0248</b>	0.0251	0.0289***	0.0391***	-
				0.00	<b>0.0378</b>	0.0817***	0.0815***	0.1084***	-
	Waxman	5	100	- 0.75	<b>0.0107</b>	0.0290***	0.0279***	0.0413***	-
				0.00	<b>0.0349</b>	0.0592***	0.0618***	0.1177***	-
		10	100	- 0.75	0.0198***	0.0152***	<b>0.0118</b>	0.0132***	-
				0.00	0.0136***	0.0092	<b>0.0091</b>	0.0189***	-
		196	100	- 0.75	0.0154***	0.0075***	<b>0.0051</b>	0.0116***	0.2852
				0.00	0.0477***	0.0152	<b>0.0149</b>	0.0403***	0.2317
196	100	- 0.75	<b>0.0076</b>	0.0160***	0.0124***	0.0191***	0.3233		
		0.00	0.0485**	<b>0.0404</b>	0.0416	0.0825***	0.4407		
				- 0.75	0.0353***	0.0165***	<b>0.0140</b>	0.0261***	-
				0.00	0.1204***	0.0995**	<b>0.0899</b>	0.1190***	-

The best values among the representations are shown in bold in each row

The values of the average Epsilon R3 quality indicator of 50 runs with each representation are shown. The results of NAMOA\* regarding the R3 quality indicator with the same time budget of 10 s are shown in the last column, for comparison. The one-sided Wilcoxon signed-rank test was used to decide statistical significance between each variant compared to the best one in each row. Results are indicated as (\*) $p < 0.05$ , (\*\*)  $p < 0.005$ , and (\*\*\*)  $p < 0.0005$

**Table 12** Comparing the best variants of NSGA-II for the four representations, separately for the 32 test instances with a time budget of 20 s

Problem instance					DirVarL-dir	DirFixL	RanKey-2ptX	IntPri-WMX	NAMOA* 20s
Obj	Graph type	$l_{\max}$	n	$\rho$					
2	Grid	5	100	- 0.75	0.0115***	0.0090	<b>0.0070</b>	0.0157***	-
				0.00	0.0265***	0.0237***	<b>0.0143</b>	0.0283***	-
				- 0.75	0.0130***	0.0088*	<b>0.0077</b>	0.0159***	-
		10	100	0.00	0.0264***	0.0316***	<b>0.0195</b>	0.0456***	-
				- 0.75	0.0154***	0.0135	<b>0.0131</b>	0.0265***	-
				0.00	0.0281	0.0338***	<b>0.0237</b>	0.0563***	-
	Waxman	5	100	- 0.75	0.0109	0.0105	<b>0.0091</b>	0.0171***	-
				0.00	<b>0.0348</b>	0.0400***	0.0382***	0.0694***	-
				- 0.75	0.0013***	<b>0.0008</b>	0.0011***	0.0009***	0.0000
		10	100	0.00	0.0083***	0.0062***	0.0032	<b>0.0030</b>	0.0000
				- 0.75	0.0137***	0.0055	<b>0.0042</b>	0.0113***	0.2190
				0.00	0.0122***	<b>0.0056</b>	0.0080***	0.0280***	0.1389
3	Grid	5	100	- 0.75	0.0071***	0.0076**	<b>0.0066</b>	0.0069***	0.2063
				0.00	0.0313***	0.0182***	<b>0.0097</b>	0.0319***	0.0000
				- 0.75	0.0112***	0.0071***	<b>0.0063</b>	0.0118***	-
		10	100	0.00	0.0234***	0.0208**	<b>0.0153</b>	0.0232***	-
				- 0.75	<b>0.0080</b>	0.0155***	0.0130***	0.0206***	-
				0.00	<b>0.0261</b>	0.0318***	0.0297***	0.0605***	-
	Waxman	5	100	- 0.75	<b>0.0110</b>	0.0174***	0.0168***	0.0257***	-
				0.00	<b>0.0280</b>	0.0561***	0.0445***	0.0779***	-
				- 0.75	0.0240	<b>0.0237</b>	0.0240***	0.0348***	-
		10	100	0.00	<b>0.0342</b>	0.0766***	0.0751***	0.0999***	-
				- 0.75	<b>0.0093</b>	0.0255***	0.0249***	0.0384***	-
				0.00	<b>0.0328</b>	0.0517***	0.0549***	0.1132***	-
Waxman	5	100	- 0.75	0.0165***	0.0151***	<b>0.0111</b>	0.0121***	-	
			0.00	0.0124***	0.0084	<b>0.0076</b>	0.0173***	0.2317	
			- 0.75	0.0116***	0.0071***	<b>0.0044</b>	0.0106***	0.2754	
	10	100	0.00	0.0456***	0.0157	<b>0.0126</b>	0.0329***	0.2317	
			- 0.75	<b>0.0063</b>	0.0165***	0.0118***	0.0176***	0.3053	
			0.00	0.0503**	0.0400	<b>0.0371</b>	0.0750***	0.3919	
196	100	- 0.75	0.0311***	0.0169***	<b>0.0120</b>	0.0233***	-		
		0.00	0.1160***	0.0938**	<b>0.0793</b>	0.1148***	-		

The best values among the representations are shown in bold in each row

The values of the average Epsilon R3 quality indicator of 50 runs with each representation are shown. The results of NAMOA\* regarding the R3 quality indicator with the same time budget of 20 s are shown in the last column, for comparison. The one-sided Wilcoxon signed-rank test was used to decide statistical significance between each variant compared to the best one in each row. Results are indicated as (\*) $p < 0.05$ , (\*\*)  $p < 0.005$ , and (\*\*\*)  $p < 0.0005$

**Acknowledgements** This work is supported in part by the Engineering and Physical Sciences Research Council (EP/N029496/1, EP/N029496/2, EP/N029356/1, EP/N029577/1, EP/N029577/2). Dr. Jun Chen is a Turing Fellow <https://www.turing.ac.uk/people/researchers/jun-chen>.

**Availability of data and materials** The data used in all presented experiments can be found at: <https://u.pcloud.link/publink/show?code=kZjt6zXZKNVrMOscFeff7e5Ou7X7auDrfwKX>.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are

included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Chen J, Weisz M, Locatelli G, Ravizza S, Atkin JA, Stewart P, Burke EK. Toward a more realistic, cost-effective, and greener ground movement through active routing: a multiobjective shortest path approach. *IEEE Trans Intell Transp Syst.* 2016;17(12):3524–40.
- Ehmke JF, Campbell AM, Thomas BW. Vehicle routing to minimize time-dependent emissions in urban areas. *Eur J Oper Res.* 2016;251(2):478–94.
- Hrnčič J, Rovatsos M, Jakob M. Ridesharing on timetabled transport services: a multiagent planning approach. *J Intell Transp Syst.* 2015;19(1):89–105.
- Qian J, Eglese R. Fuel emissions optimization in vehicle routing problems with time-varying speeds. *Eur J Oper Res.* 2016;248(3):840–8.
- Wu X, He X, Yu G, Harmandayan A, Wang Y. Energy-optimal speed control for electric vehicles on signalized arterials. *IEEE Trans Intell Transp Syst.* 2015;16(5):2786–96.
- Garaix T, Artigues C, Feillet D, Josselin D. Vehicle routing problems with alternative paths: an application to on-demand transportation. *Eur J Oper Res.* 2010;204(1):62–75.
- Serafini P. Some considerations about computational complexity for multi objective combinatorial problems. In: Johannes J, Werner K (eds) Recent advances and historical development of vector optimization. Springer; 1987. p. 222–32.
- Chitra C, Subbaraj P. A nondominated sorting genetic algorithm solution for shortest path routing problem in computer networks. *Expert Syst Appl.* 2012;39(1):1518–25.
- Li R, Leung Y, Huang B, Lin H. A genetic algorithm for multi-objective dangerous goods route planning. *Int J Geogr Inf Sci.* 2013;27(6):1073–89.
- Lin L, Gen M. An effective evolutionary approach for bicriteria shortest path routing problems. *IEEJ Trans Electron Inf Syst.* 2008;128(3):416–23.
- Blum C, Roli A. Metaheuristics in combinatorial optimization: overview and conceptual comparison. *ACM Comput Surv (CSUR).* 2003;35(3):268–308.
- Abbaspour RA, Samadzadegan F. An evolutionary solution for multimodal shortest path problem in metropolises. *Comput Sci Inf Syst.* 2010;7(4):789–811.
- Yu H, Lu F. A multi-modal route planning approach with an improved genetic algorithm. In: Wenzhong S, Michael G, Brian L, Yee L (eds) Advances in geo-spatial information science; 2012. pp 193–202 ISBN 978-0-415-62093-2. [https://books.google.co.uk/books?hl=en&lr=&id=CAi6FKV00SYC&oi=fnd&pg=PA193&ots=qYwVLiIkwL&sig=JkDe3kB0m0vbasyp\\_X6Ch9tu4XE#v=onepage&q&f=false](https://books.google.co.uk/books?hl=en&lr=&id=CAi6FKV00SYC&oi=fnd&pg=PA193&ots=qYwVLiIkwL&sig=JkDe3kB0m0vbasyp_X6Ch9tu4XE#v=onepage&q&f=false).
- Ahn CW, Ramakrishna RS. A genetic algorithm for shortest path routing problem and the sizing of populations. *IEEE Trans Evol Comput.* 2002;6(6):566–79.
- Inagaki J, Haseyama M, Kitajima H. A genetic algorithm for determining multiple routes and its applications. In: ISCAS'99. Proceedings of the 1999 IEEE international symposium on circuits and systems VLSI (Cat. No. 99CH36349), vol. 6. IEEE; 1999. p. 137–40.
- Gen M, Altıparmak F, Lin L. A genetic algorithm for two-stage transportation problem using priority-based encoding. *OR Spectr.* 2006;28(3):337–54.
- Burke EK, Newall JP, Wear RF. Initialization strategies and diversity in evolutionary timetabling. *Evol Comput.* 1998;6(1):81–103.
- Deng Y, Liu Y, Zhou D. An improved genetic algorithm with initial population strategy for symmetric tsp. *Mathemat Prob Eng.* 2015;1–6. <https://doi.org/10.1155/2015/212794>
- Hasan BS, Khamees MA, Mahmoud ASH. A heuristic genetic algorithm for the single source shortest path problem. In: 2007 IEEE/ACS international conference on computer systems and applications. IEEE; 2007. p. 187–94.
- Lee J, Kim DW. An effective initialization method for genetic algorithm-based robot path planning using a directed acyclic graph. *Inf Sci.* 2016;332:1–18.
- Beke L, Weisz M, Chen J. A comparison of genetic representations for multi-objective shortest path problems on multigraphs. In: European conference on evolutionary computation in combinatorial optimization (part of EvoStar). Springer; 2020. p. 35–50.
- Munetomo M, Takai Y, Sato Y. An adaptive network routing algorithm employing path genetic operators. In: ICGA 1997.
- Ji Z, Kim YS, Chen A. Multi-objective  $\alpha$ -reliable path finding in stochastic networks with correlated link costs: a simulation-based multi-objective genetic algorithm approach (smoga). *Expert Syst Appl.* 2011;38(3):1515–28.
- Lin L, Gen M. Priority-based genetic algorithm for shortest path routing problem in ospf. In: Intelligent and evolutionary systems. Springer; 2009. p. 91–103.
- Mohammed AW, Sahoo NC, Geok TK. Solving shortest path problem using particle swarm optimization. *Appl Soft Comput.* 2008;8(4):1643–53.
- Gen M, Cheng R, Wang D. Genetic algorithms for solving shortest path problems. In: Proceedings of 1997 IEEE international conference on evolutionary computation (ICEC'97). IEEE; 1997. p. 401–6.
- Gen M, Lin L. A new approach for shortest path routing problem by random key-based ga. In: Proceedings of the 8th annual conference on genetic and evolutionary computation. ACM; 2006. p. 1411–2.
- Bean JC. Genetic algorithms and random keys for sequencing and optimization. *ORSA J Comput.* 1994;6(2):154–60.
- Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Trans Evol Comput.* 2002;6(2):182–97.
- Seada H, Deb K. U-nsga-iii: a unified evolutionary algorithm for single, multiple, and many-objective optimization. COIN report. 2014;2014022.
- This research utilised queen mary's apocrita hpc facility, supported by qmul research-it. <https://doi.org/10.5281/zenodo.438045>
- Inspyred: Bio-inspired algorithms in python. <https://www.pytho.nhosted.org/inspyred/>. Accessed 30 Oct 2019.
- López-Ibáñez M, Dubois-Lacoste J, Pérez Cáceres L, Stützle T, Birattari M. The irace package: Iterated racing for automatic algorithm configuration. *Oper Res Perspect.* 2016;3, 43–58. <https://doi.org/10.1016/j.orp.2016.09.002>.
- Waxman BM. Routing of multipoint connections. *IEEE J Sel Areas Commun.* 1988;6(9):1617–22.
- Machuca E, Mandow L, de la Cruz JLP, Ruiz-Sepulveda A. An empirical comparison of some multiobjective graph search algorithms. In: Annual conference on artificial intelligence. Springer; 2010. p. 238–45.
- Mote J, Murthy I, Olson DL. A parametric approach to solving bicriterion shortest path problems. *Eur J Oper Res.* 1991;53(1):81–92.

37. Mandow L, De la Cruz JP, et al. A new approach to multiobjective a\* search. In: IJCAI, vol. 8. Citeseer; 2005.
38. Tung CT, Chew KL. A multicriteria pareto-optimal path algorithm. *Eur J Oper Res.* 1992;62(2):203–9.
39. Liefoghe A, Derbel B. A correlation analysis of set quality indicator values in multiobjective optimization. *Proc Genet Evol Comput Conf.* 2016;2016:581–8.
40. Fonseca CM, Knowles JD, Thiele L, Zitzler E. A tutorial on the performance assessment of stochastic multiobjective optimizers. In: Third international conference on evolutionary multi-criterion optimization (EMO 2005), vol. 216; 2005. p. 240.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.