

# Stochastic Pareto local search: Pareto neighbourhood exploration and perturbation strategies

Mădălina M. Drugan · Dirk Thierens

Received: 9 March 2011 / Accepted: 25 June 2012 / Published online: 3 July 2012  
© The Author(s) 2012. This article is published with open access at Springerlink.com

**Abstract** Pareto local search (PLS) methods are local search algorithms for multi-objective combinatorial optimization problems based on the Pareto dominance criterion. PLS explores the Pareto neighbourhood of a set of non-dominated solutions until it reaches a local optimal Pareto front. In this paper, we discuss and analyse three different Pareto neighbourhood exploration strategies: best, first, and neutral improvement. Furthermore, we introduce a deactivation mechanism that restarts PLS from an archive of solutions rather than from a single solution in order to avoid the exploration of already explored regions. To escape from a local optimal solution set we apply stochastic perturbation strategies, leading to stochastic Pareto local search algorithms (SPLS). We consider two perturbation strategies: mutation and path-guided mutation. While the former is unbiased, the latter is biased towards preserving common substructures between 2 solutions. We apply SPLS on a set of large, correlated bi-objective quadratic assignment problems (bQAPs) and observe that SPLS significantly outperforms multi-start PLS. We investigate the reason of this performance gain by studying the fitness landscape structure of the bQAPs using random walks. The best performing method uses the stochastic perturbation algorithm, the first improvement Pareto neighborhood exploration and the deactivation technique.

---

M.M. Drugan (✉) · D. Thierens  
Department of Information and Computing Sciences, Universiteit Utrecht, P.O. Box 80.089,  
3508TB Utrecht, The Netherlands  
e-mail: [mdrugan@vub.ac.be](mailto:mdrugan@vub.ac.be)

D. Thierens  
e-mail: [D.Thierens@uu.nl](mailto:D.Thierens@uu.nl)

M.M. Drugan  
Computational Modeling Lab, Department of Computer Science, Vrije Universiteit Brussel,  
Pleinlaan 2, 1050 Brussels, Belgium

**Keywords** Stochastic Pareto local search · Pareto neighbourhood improvement strategies · Pareto neighbourhood perturbation strategies · Path-guided mutation

## 1 Introduction

*Pareto local search algorithms* (PLS) (Paquete et al. 2004; Angel et al. 2004; Basseur 2006) are local—or neighbourhood—search techniques for solving multi-objective combinatorial optimization problems. PLS applies an exploration strategy to iteratively move from a set of current solutions to a neighbouring set that improves upon the current solution. The algorithm stops in a *Pareto local optimal set*: a set of solutions that have no improving solutions in their neighbourhood. Because the local search gets stuck in a local optimal set, the search needs to be restarted from different regions of the search space. Multi-start PLS restarts the search from randomly generated points in the search space. However, this does not exploit information present in previously found solutions.

In the literature some variants of PLS algorithms can be found, each having small differences in their neighbourhood exploration technique. Paquete et al.'s PLS (Paquete et al. 2004) stores only those solutions in the archive that are Pareto optimal. Basseur's PLS (Basseur 2006) explores *all* solutions in the archive exactly once, even though they might have become dominated by newly included solutions. Both PLSs explore the entire neighbourhood of solutions in the archive. Thus, these PLSs use a *best improvement exploration strategy* (Hansen and Mladenović 2006). Angel et al.'s PLS (Angel et al. 2004) uses dynasearch and dynamic programming to explore the neighbourhood of the bi-objective travelling salesman problem. Liefvooghe et al. (2011) experimentally compare various instances of PLS algorithms using different parameters and settings on bi-objective instances of travelling salesman and scheduling problems.

Variations of PLS were proposed to improve PLS performance (Hamacher and Ruhe 1994). Two phase PLS (Lust and Teghem 2010; Dubois-Lacoste et al. 2011a) first approximates the Pareto local optimal set with solutions generated with single-objective solvers. In the second phase, the algorithm uses PLS (Paquete et al. 2004; Angel et al. 2004) to generate solutions not found in the first phase. Guided PLS (Alsheddy and Tsang 2009) uses a variant of PLS that prioritizes the objectives of the search space to uniformly spread the solutions over the Pareto local optimal set. Guided PLS is somehow the inverse of the two phase PLS because it first performs PLS and then tries to escape from a Pareto local optimal set by adjusting the weights of different objectives.

Iterated PLS (Drugan and Thierens 2010) uses stochastic perturbation operators to escape from a Pareto local optimal set. Iterated PLS is the multi-objective equivalent of iterated local search (Lourenco et al. 2003) a technique for single objective optimization that restarts LS from solutions generated by mutating the currently found local optima. In addition to mutation SPLS can generate restarting solutions using recombination operators because there are multiple solutions in the Pareto local optimal set that can be used as parent solutions. The Pareto local optimal set can be used directly as the population or alternatively, a separate elitist population can be used to generate new solutions (Lopez-Ibanez et al. 2006).

*Main contributions of the paper* We propose several techniques to ameliorate the performance of Pareto local search algorithms. Our variation of PLS uses a technique—called *deactivation*—that restarts PLS from a newly generated solution and a set of incomparable solutions that were generated with previously restarted PLS runs. In Sect. 2, we show that PLS using deactivation can avoid the repetitive exploration of (non-promising) regions.

PLS algorithms usually explore the entire neighbourhood of a solution (Paquete et al. 2004). We show here though that first improvement neighbourhood exploration strategies can be more efficient. A similar observation has recently been made in (Liefvooghe et al. 2011). *Pareto neighbourhood exploration strategies*—or improvement strategies—are the equivalent of single-objective improvement strategies (Hansen and Mladenović 2006) for multi-objective spaces. We view Pareto improvement as a relationship between an initial solution, a solution from the neighbourhood of the initial solution, and all the other solutions of the current Pareto set. In Sect. 3, we design and analyse three Pareto neighbourhood exploration strategies. We show that these Pareto improvement strategies are well performing. We prove that PLS using our Pareto neighbourhood exploration strategies are proper, sound and complete. We also discuss the relationship between our Pareto improvement strategies and another, often used, technique to search in multi-objective search spaces, namely the hypervolume unary indicator (Zitzler et al. 2003; Bringmann and Friedrich 2010). The (Pareto) improvement strategies are independent of the landscape they are applied on, thus these results can straightforwardly be applied to other combinatorial problems.

To escape from a local optimal solution set we apply stochastic perturbation strategies, leading to *stochastic Pareto local search algorithms* (SPLS). We consider two perturbation strategies: mutation and path-guided mutation. While the former is unbiased, the latter is biased towards preserving common substructures between two solutions. In Sect. 4 we discuss a SPLS that uses mutation and path-guided mutation as perturbation operators. Since path-guided mutation can be seen as a specific form of recombination we refer to this algorithm as Genetic PLS (GPLS). The structure of the neighbourhood and the perturbation operators depend on the representation and particularities of the problem they are tested on. In Sect. 5, we experimentally investigate the efficiency of the proposed mechanisms (i.e., the Pareto improvement strategies, deactivation, genetic operators) for PLSs on instances of the bi-objective QAP (bQAP) (Knowles and Corne 2003). The solutions of bQAPs are represented by permutations of facilities to different locations.

The landscape analysis in Sect. 6 records the quality of the restarting solutions with random walks (RW). We extend the scope of the RW method to study the relationship between the distance to a Pareto local optimal set and quality measures, like the hypervolume unary indicator, or mechanisms like deactivation. RW is a logical choice for analysing the behaviour of GPLS in relationship with the genetic operators because the solutions generated with these operators are *all* accepted as restarting points for PLS (Merz and Freisleben 2000). RW reveals correlations between the structure of the search space and the genetic operator used. Section 7 concludes the paper.

## 2 Stochastic Pareto local search (SPLS)

We define stochastic Pareto local search (SPLS) algorithms as a combination of stochastic perturbation operators and Pareto local search (PLS). In Sect. 2.1, we first introduce a slight generalization of PLS (Paquete et al. 2004; Angel et al. 2004; Basseur 2006) to allow the design of more efficient SPLS algorithms. In Sect. 2.3, we present the genetic PLS algorithms that use genetic operators to restart PLS.

*Notation and definitions* Let us consider the multi-objective function  $\mathbf{f} = (f_1, \dots, f_m)$ , where  $\mathbf{f} : \mathcal{S} \rightarrow \mathcal{O}$ .  $\mathcal{S}$  is the set of solutions in a countable *solution space*. The image of the solution set  $\mathcal{S}$  under  $\mathbf{f}$  is denoted with the *objective space*  $\mathcal{O} = \mathbf{f}(\mathcal{S}) = \{y \mid \exists x \in \mathcal{S} : y = \mathbf{f}(x)\}$ . We consider the single objective space as a special case of the multi-objective space with  $m = 1$ . A bi-objective space has  $m = 2$ .

Let  $a, b \in \mathcal{S}$  be two solutions and  $\mathbf{f}(a), \mathbf{f}(b) \in \mathcal{O}$  be their images in objective space, called objective vectors. To define an optimization problem, we first consider an order binary relationship,  $\preceq$ , between any two elements in the objective space  $\mathcal{O}$ . Without loss of generality, only minimization is considered. In Table 1, we present the relationships between objective vectors and sets of objective vectors used in this paper (Zitzler et al. 2003).

A *Pareto set* is a set of non-dominated solutions. The set of *Pareto optimal solutions*, or *Pareto optimal set*,  $\mathcal{A} \subset \mathcal{S}$ , is  $\text{Min}(\mathcal{S}, <) = \{a \in \mathcal{A} \subset \mathcal{S} \mid \neg \exists x \in \mathcal{S} : \mathbf{f}(x) < \mathbf{f}(a)\}$ . A *Pareto local optimal set* is the set of non-dominated solutions  $\mathcal{A}_{alg}$  generated with a search algorithm  $alg$ ,  $\text{Min}(\mathcal{S}_{alg}, <) = \{a \in \mathcal{A}_{alg} \subset \mathcal{S}_{alg} \subset \mathcal{S} \mid \neg \exists x \in \mathcal{S}_{alg} : \mathbf{f}(x) < \mathbf{f}(a)\}$ , where  $\mathcal{S}_{alg} \subset \mathcal{S}$  is the subset of solutions explored with the search algorithm. A *maximal Pareto local set* is a Pareto local optimal set containing solutions for which the solutions in their neighbourhood are either dominated or in the Pareto local optimal set.

### 2.1 Pareto local search

Algorithm 1 presents the pseudo-code of our PLS algorithm. For ease of discussion, we consider an archive of unlimited size that can store *all* the solutions in the Pareto set. For very large Pareto sets, for instance for real coded problems where unlimited

**Table 1** Relations between objective vectors and sets of objective vectors considered in this paper

Relation	Objective vectors		Sets of objective vectors	
Dominance	$\mathbf{f}(a) < \mathbf{f}(b)$	$\exists i, f_i(a) < f_i(b)$ and $\forall j \neq i, f_j(a) \leq f_j(b)$	$A < B$	$\forall \mathbf{f}(b) \in B, \exists \mathbf{f}(a) \in A$ such that $\mathbf{f}(a) < \mathbf{f}(b)$
Weakly dominates	$\mathbf{f}(a) \preceq \mathbf{f}(b)$	$\forall i, f_i(a) \leq f_i(b)$	$A \preceq B$	$\forall \mathbf{f}(b) \in B, \exists \mathbf{f}(a) \in A$ such that $\mathbf{f}(a) \preceq \mathbf{f}(b)$
Incomparable	$\mathbf{f}(a) \parallel \mathbf{f}(b)$	$\mathbf{f}(a) \not\preceq \mathbf{f}(b)$ and $\mathbf{f}(b) \not\preceq \mathbf{f}(a)$	$A \parallel B$	$A \not\preceq B$ and $B \not\preceq A$
Non-dominated by	$\mathbf{f}(a) \not\prec \mathbf{f}(b)$	$\mathbf{f}(a) \leq \mathbf{f}(b)$ or $\mathbf{f}(a) \parallel \mathbf{f}(b)$	$A \not\prec B$	$A \preceq B$ or $A \parallel B$

---

**Algorithm 1** Pareto local search  $PLS(A, \mathcal{I})$

---

**Require:**  $\exists s \in A, s.visited = false$   
 $A' \leftarrow A$   
**while**  $\exists s \in A',$  with  $s.visited = false$  **do**  
 $A'' \leftarrow \mathcal{I}(s, A' \setminus \{s\}, \mathcal{N}, \leq, \mathbf{f})$   
**for all**  $s' \in A''$  **do**  
 $s'.visited \leftarrow false$   
**end for**  
 $A' \leftarrow \text{merge}(A', A'')$   
Set  $s.visited \leftarrow true$   
**end while**  
**return**  $A'$

---

size Pareto sets are unrealistic, different archiving methods can be applied (Knowles and Corne 2004; Deb et al. 2005; Lopez-Ibanez et al. 2011).

$PLS(A, \mathcal{I})$  has two input parameters.  $A$  is a Pareto set containing at least one solution  $s$  with visited flag set to *false*. The Pareto improvement strategy,  $\mathcal{I}(s, A' \setminus \{s\}, \mathcal{N}, \leq, \mathbf{f})$ , explores the neighbourhood  $\mathcal{N}$  of a solution  $s$  using the Pareto set  $A' \setminus \{s\}$  and the domination relationship  $\leq$  over the function  $\mathbf{f}$ . The output of  $\mathcal{I}(\cdot)$  is a Pareto set,  $A''$ . The neighbourhood function  $\mathcal{N} : \mathcal{S} \rightarrow \mathcal{P}(\mathcal{S})$  has as input a solution from  $\mathcal{S}$  and returns the set of neighbours for that solution.  $\mathcal{N}(s)$  depends on the problem (e.g., multi-objective QAPs) and the solution  $s$  it is applied on.  $\mathcal{I}$ 's algorithm does not depend on the landscape; it either exhaustively explores the entire neighbourhood and selects all non-dominated solutions in it, or stops when the first improvement to the initial set of solutions is found.

Each iteration of PLS, a solution  $s$  with the visited flag set to false,  $s.visited = false$ , is randomly chosen from the current Pareto set (or archive)  $A$ . The solutions in the neighbourhood of  $s$  are evaluated with  $\mathcal{I}$ . The function *merge* merges  $M \geq 2$  Pareto sets into a new Pareto set, where

$$\begin{aligned} &\text{merge}(A_1, \dots, A_M) \\ &= \{s \in A_1 \vee \dots \vee s \in A_M \mid \neg \exists (s' \in A_1 \vee \dots \vee s' \in A_M) : \mathbf{f}(s') < \mathbf{f}(s)\} \end{aligned}$$

Note that *merge* also removes the dominated solutions from the input Pareto set. The newly added solutions from the neighbourhood have the visited flag set on “false”. The search continues until there are no unvisited solutions in  $A'$ . Thus, the search stops when there are no “new” incomparable or dominating solutions found.

*Deactivation technique* We use a Pareto set  $A$  to restart PLS in Algorithm 1 as opposed to a single initial solution as Paquete et al.’s PLS does.

**Definition 1** Let’s consider the deactivation method from Algorithm 2,  $Deactiv(s, A)$ , with  $A$  a Pareto set and  $s$  a solution. The output is an archive  $A_I$  containing  $s$  and the solutions of  $A$  that are incomparable with  $s$ .

The differences between PLS from Algorithm 1 and Paquete et al.’s PLS are: (i) we use a Pareto set  $A$  to restart PLS, and (ii) we propose several instances for the Pareto

**Algorithm 2** Deactivation Deactiv( $s, A$ )**Require:**  $A$ : a Pareto set

---

```

 $A_I \leftarrow \{s\}$ 
for all  $s' \in A$  do
  if  $\mathbf{f}(s) \parallel \mathbf{f}(s')$  then
     $A_I \leftarrow \text{merge}(\{s'\}, A_I)$ 
  end if
end for
return  $A_I$ 

```

---

improvement algorithm  $\mathcal{I}$ . PLS restarting from a single solution  $s$  and using the best improvement neighbourhood exploration strategy resembles the most Paquete et al.'s PLS. The following proposition shows that deactivation can decrease the number of visited solutions, the extra solutions visited with PLS without deactivation that are dominated by solutions in the deactivation set.

**Proposition 1** *Let PLS be as in Algorithm 1, with  $\mathcal{I}$  the best Pareto improvement strategy. Consider two initial Pareto sets  $A_1 = \{s\}$  and  $A_2 = \{s, q_1, \dots, q_k\}$  for PLS. Consider the same sequence  $\pi_{\mathcal{N}(s)}$  of visiting the solutions in  $\mathcal{N}(s)$ . Then, the number of solutions from  $\mathcal{N}(s)$  visited with  $\text{PLS}(A_1, \mathcal{I})$  is smaller or equal with the number of solutions  $\mathcal{N}(s)$  visited with  $\text{PLS}(A_2, \mathcal{I})$ . The set of solutions in  $\mathcal{N}(s)$  visited only with  $\text{PLS}(A_1, \mathcal{I})$  are dominated by at least one solution from  $\{q_1, \dots, q_k\}$ .*

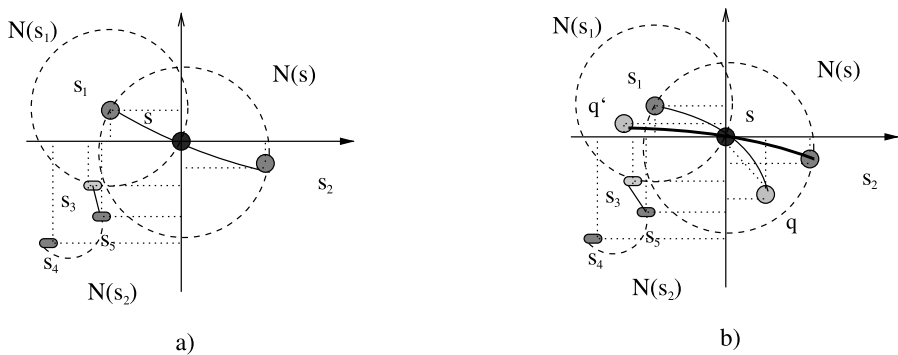
*Proof* Without loss of generality, let's assume that  $k = 1$ . Then,  $A_2 = \{s, q_1\}$ . Consider the sequence of solutions  $\{s_1, \dots, s_n\} \in \pi_{\mathcal{N}(s)}$  and these solutions' relationships with  $s$  and  $q_1$ . The only case when the two PLSs are behaving differently is when  $\exists s_i$  that is dominated by  $q_1$  and incomparable with  $s$ . With  $\text{PLS}(A_1, \mathcal{I})$ ,  $s_i$  is added to  $A'$ . With  $\text{PLS}(A_2, \mathcal{I})$ ,  $s_i$  is not added to  $A'$ . We conclude that there are less solutions from  $\pi_{\mathcal{N}(s)}$  that are evaluated with PLS using  $A_2$  than when  $A_1$  is used, and the extra solutions evaluated with  $A_1$  are dominated by  $q_1$ .  $\square$

Example 1 shows situations where the deactivation techniques is useful.

*Example 1* In Fig. 1, a solution is denoted with a point. A circle or a part of a circle denotes the neighbourhood of a solution. A Pareto set is denoted with a curve. In Fig. 1, the initial solution  $s$  has in its neighbourhood  $\mathcal{N}(s)$  two incomparable solutions  $s_1$  and  $s_2$ ,  $\mathbf{f}(s_1) \parallel \mathbf{f}(s_2)$ ,  $\mathbf{f}(s) \parallel \mathbf{f}(s_2)$  and  $\mathbf{f}(s) \parallel \mathbf{f}(s_1)$ , where  $s_1, s_2 \in \mathcal{N}(s)$ . Let  $s_3$  be in the neighbourhood of  $s_1$ ,  $s_3 \in \mathcal{N}(s_1)$ , such that  $s_3$  dominates  $s$ ,  $s_1$  and  $s_2$ ,  $\mathbf{f}(s_3) < \mathbf{f}(s)$  and  $\mathbf{f}(s_3) < \mathbf{f}(s_1)$  and  $\mathbf{f}(s_3) < \mathbf{f}(s_2)$ . Let  $s_4$  and  $s_5$  be in the neighbourhood of  $s_2$ ,  $s_4, s_5 \in \mathcal{N}(s_2)$ , such that  $s_4$  dominates *all* other solutions, and  $s_5$  is incomparable with  $s_3$  and dominates  $s_1$ ,  $s_2$  and  $s$ .

Let's consider PLS,  $\text{PLS}(A, \mathcal{I})$ , from Algorithm 1 with  $\mathcal{I}$  the best Pareto improvement strategy, and different Pareto sets  $A$ .

In Fig. 1(a), we consider  $A = \{s\}$  and two equal probable sequences to visit the neighbourhood of  $s$ ,  $\{s_1, s_2\}$  and  $\{s_2, s_1\}$ . When the neighbours of  $s$  are visited in the



**Fig. 1** Comparing the sequence of visited solutions for (a)  $PLS(s, \mathcal{I})$  and (b)  $PLS(\{s, q\}, \mathcal{I})$  or  $PLS(\{s, q'\}, \mathcal{I})$

---

**Algorithm 3** Multi-restart PLS MPLS( $\mathcal{I}, \mathcal{T}$ )

---

```

A ← ∅
while Stopping criterion,  $\mathcal{T}$ , is NOT met do
    Generate  $s$  uniform randomly
     $A' \leftarrow Deactiv(s, A)$ 
     $A \leftarrow merge(A, PLS(A', \mathcal{I}))$ 
end while
return A
    
```

---

sequence  $\{s_1, s_2\}$ , the PLS's output is  $A' = \{s_3\}$  because  $s_3 \in \mathcal{N}(s_1)$  is dominating  $s_2$  that is deleted from the archive  $A'$ . When the neighbours of  $s$  are visited in the sequence  $\{s_2, s_1\}$ , the PLS's output is  $A' = \{s_4\}$  because  $s_3 \in \mathcal{N}(s_1)$  is added to the archive  $A'$  after adding  $s_4 \in \mathcal{N}(s_2)$  and  $s_4$  dominates  $s_3$ .

In Fig. 1(b),  $q$  and  $q'$  are two solutions that are incomparable with  $s$ , and  $q$  dominates  $s_2$  and is incomparable with  $s_1$ , and  $q'$  dominates  $s_1$  and is incomparable with  $s_2$ . When  $A = \{s, q\}$ , the PLS's output is  $A' = \{s_3\}$  because  $s_2$  is dominated by  $q$  and is deleted from the archive  $A'$ . When  $A = \{s, q'\}$ , then  $s_1$  is dominated by  $q'$  and is deleted from the archive  $A'$ , and the output is  $A' = \{s_4\}$ .

Note that both PLSs can delete  $s_3$  whose neighbourhood contains  $s_4$  that dominates all the other solutions. In the first case, the output of PLS depends on the sequence in which solutions in the neighbourhood of  $s$  are evaluated. In the second case, the output of PLS depends on the initial Pareto set.

In Algorithms 3 and 4, the input Pareto set  $A$  of PLS is equal to the output of the deactivation method.

2.2 The multi-restart PLS algorithm (MPLS)

MPLS is a straightforward algorithm to escape from a Pareto local optimal set. It restarts multiple times from uniform randomly chosen initial solutions. Algorithm 3 presents the pseudo-code for MPLS. The input parameters for MPLS are a Pareto

**Algorithm 4** Genetic PLS GPLS( $\mathcal{I}, \alpha, \mathcal{T}_1, \mathcal{T}$ )

---

```

A ← MPLS( $\mathcal{I}, \mathcal{T}_1$ )
while Stopping criterion,  $\mathcal{T}$ , is NOT met do
  Select  $s$  uniform randomly from  $A$ 
  if  $\alpha > U(0, 1)$  or  $|A| < 2$  then
     $s' \leftarrow$  Mutation( $s$ )
  else
    Select  $s_1 \neq s$  from  $A$ 
     $s' \leftarrow$  Recombine( $s, s_1$ )
  end if
   $A' \leftarrow$  Deactiv( $s', A$ )
   $A \leftarrow$  merge( $A, \text{PLS}(A', \mathcal{I})$ )
end while
return  $A$ 

```

---

improvement strategy  $\mathcal{I}$  and a stopping criterion  $\mathcal{T}$ . The stopping criterion can be, for example, a maximum number of restarted PLSs. The Pareto set  $A$  is initialized as the empty set,  $A \leftarrow \emptyset$ . Until  $\mathcal{T}$  is met, a solution  $s$  is uniform randomly sampled from the search space. The dominating solutions in  $A$  by  $s$  are deactivated,  $A' \leftarrow \text{Deactiv}(s, A)$ . The output Pareto set  $A'$  is the input Pareto set for PLS,  $\text{PLS}(A', \mathcal{I})$ . The Pareto set  $A$  is updated by merging it with the output of PLS.

The difference between this multi-start PLS and Paquete's multi-start PLS is: (i) the use of the deactivation technique, and (ii) the use of different Pareto improvement strategies  $\mathcal{I}$ . Paquete et al.'s PLS starts to explore the neighborhood of a solution with an empty archive. MLS cannot really profit from an exploration archive, because the restarting solution  $s$  is uniform randomly generated, it is improbable that there are solutions in  $A'$  beside  $s$ . That is, all the solutions in the Pareto local optimum set  $A$  are probably dominating a randomly generated solution  $s$ . The deactivation technique will lead to an empty archive. This hypothesis is confirmed in Sect. 6, where MPLS from Algorithm 3 and Paquete's MPLS have similar performance. Therefore, deactivation is not useful unless fit restarting solutions are generated, for example using genetic operators on solutions from the Pareto local optimum set  $A$ . We show that the use of this exploration archive is very useful for GPLS.

### 2.3 Restarting PLS using genetic operators

To improve upon multi-restart PLS's performance, stochastic Pareto local search aims to escape from Pareto local optimal sets by stochastic perturbation operators that preserve partial information of the perturbed solutions. In case of *genetic PLS* (GPLS), the perturbation operators are a combination of mutation and recombination. Algorithm 4 gives the pseudo-code of GPLS.

First, as initialization phase, an MPLS runs in order to construct an initial Pareto set  $A$  with diverse solutions from different basins of attraction. The stopping criterion  $\mathcal{T}_1$  is the number of times  $N > 0$  MPLS is restarted from uniform randomly generated solutions. The type of operator, i.e. mutation or recombination, is selected proportionally with a fixed probability  $\alpha \in [0, 1]$ . A new individual  $s'$  is generated



with mutation,  $\text{Mutation}(s)$ , or with recombination,  $\text{Recombination}(s, s_1)$ , where  $s_1$  is another individual from  $A$ ,  $s \neq s_1$ . The solutions in  $A$  are deactivated with  $\text{Deactiv}(s', A)$  and the resulting Pareto set  $A'$  is the initial Pareto set for PLS. PLS is restarted from  $s'$ . The archive  $A$  is updated with the output of PLS,  $A''$ . This process is repeated until a stopping criterion,  $\mathcal{T}$ , is met.  $\mathcal{T}$  depends on the problem, e.g., mQAPs, and is presented in the next section. The GPLS algorithm returns a Pareto local optimal set.

When  $\alpha = 1$  the individuals are generated only with mutation. The advantage of mutation-generated restarts is that the local search is restarted from nearby areas of the landscape. When  $\alpha = 0.5$  the individuals are generated 50 % with mutation and 50 % with recombination. Recombination generates solutions at larger distance from the current solution but still aims to preserve partial information of the two parent solutions.

Note that the deactivation method does not have any effect if there are no solutions in the Pareto set  $A$  that are incomparable with the solution  $s$ . We assume that the individuals generated with mutation or recombination from solutions in the current Pareto set are “good” solutions that are incomparable with some solutions in the Pareto set.

### 3 Pareto neighbourhood exploration strategies

The performance of LS algorithms depends on the choice of improvement strategy (Hansen and Mladenović 2006; Liefoghe et al. 2011). Until recently, best improvement algorithms were commonly used for the neighbourhood exploration strategies in Pareto local search algorithms. The *first non-dominating neighbour* (Liefoghe et al. 2011) stops after the first non-dominating solution found. The first dominating neighbour (Aguirre and Tanaka 2005; Liefoghe et al. 2011; Dubois-Lacoste et al. 2011b) stops after the first dominating solution.

In this section, we introduce a new concept for neighbourhood exploration strategies that compares the solutions in a neighbourhood  $\mathcal{N}(s)$  not only to  $s$  but also to the solutions from the Pareto set  $A$ . In Algorithm 1,  $A$  is a Pareto set of solutions incomparable with  $s$  that were previously generated in a PLS run. We show that usage of  $A$  can: (i) improve the quality of the outputted Pareto set, and (ii) return a smaller Pareto set. We compare our Pareto improvement strategies with another very important method to search through the multi-objective space, the hypervolume unary indicator.

Because we consider one type of neighbourhood function,  $\mathcal{N}$ , one type of order relationship  $\preceq$ , and one function  $\mathbf{f}$ , we will often use  $\mathcal{I}(s, A)$  instead of  $\mathcal{I}(s, A, \mathcal{N}, \preceq, \mathbf{f})$  to shorten the notation.

#### 3.1 Best Pareto improvement, $\mathcal{I}_B$

Like the best improvement for single objective spaces, the best Pareto improvement strategy explores *all* the solutions in the neighbourhood of the input solution  $s$ . The pseudo-code for  $\mathcal{I}_B$  is given in Algorithm 5.

**Algorithm 5** Best Pareto Improvement  $\mathcal{I}_B(s, A, \mathcal{N}, \preceq, \mathbf{f})$ 


---

```

 $A' \leftarrow \{s\} \cup A$ 
for all  $s' \in \mathcal{N}(s)$  do
  if  $\forall s'' \in A', \mathbf{f}(s') \prec \mathbf{f}(s'') \vee \mathbf{f}(s') \parallel \mathbf{f}(s'')$  then
     $s'.visited \leftarrow false$ 
     $A' \leftarrow \text{merge}(A', \{s'\})$ 
  end if
end for
 $A' \leftarrow A' \setminus (\{s\} \cup A)$ 
Return  $A'$ 

```

---

**Definition 2** The *best Pareto improvement* strategy from Algorithm 5,  $\mathcal{I}_B(s, A)$ , evaluates *all* the individuals in the neighbourhood of  $s$ ,  $\mathcal{N}(s)$ . The outputted set  $A'$  is empty if there are no solutions in  $\mathcal{N}(s)$  that are incomparable or dominating all solutions in  $\{s\} \cup A$ . Otherwise,  $A'$  contains solutions from  $\mathcal{N}(s)$  that are dominating or incomparable with  $\{s\} \cup A$  and all other solutions from  $\mathcal{N}(s)$ . The visited flag of newly added solutions in  $A'$  is set to *false*.

Note that the output  $A'$  does not contain solutions from the initial set  $\{s\} \cup A$ . This initial set is eliminated in the last line of the algorithm  $\mathcal{I}_B$ ,  $A' \leftarrow A' \setminus (\{s\} \cup A)$ . In the next proposition, we show that the difference between the outputs of the standard best improvement strategy,  $\mathcal{I}_B(s, \emptyset)$ , and  $\mathcal{I}_B(s, A)$  is the size of the outputted Pareto set  $A'$ . The output of  $\mathcal{I}_B(s, \emptyset)$  could be larger than  $\mathcal{I}_B(s, A)$ , where the differences are solutions that are dominated by at least one solution from  $A$ . We also show that merging the resulting Pareto set of the two  $\mathcal{I}_B$  with  $A$  leads to the same result.

**Proposition 2** Let  $\mathcal{I}_B$  be as in Definition 2. Then, between the Pareto sets  $\mathcal{I}_B(s, A)$  and  $\mathcal{I}_B(s, \emptyset)$  are the following relationships:

1.  $\mathcal{I}_B(s, A) \subseteq \mathcal{I}_B(s, \emptyset)$ , and  $\forall s' \in \mathcal{I}_B(s, \emptyset) \setminus \mathcal{I}_B(s, A)$ ,  $\exists s'' \in A$ , such that  $\mathbf{f}(s'') \prec \mathbf{f}(s')$
2.  $\text{merge}(\mathcal{I}_B(s, \emptyset), A) = \text{merge}(\mathcal{I}_B(s, A), A)$

*Proof* The first relationship holds because there can be solutions in  $\mathcal{N}(s)$  that are incomparable with  $s$  but dominated by a solution from  $A$ . Then, the output of  $\mathcal{I}_B(s, \emptyset)$  contains such a solution, but  $\mathcal{I}_B(s, A)$  does not include it. In all other cases, the output of the two improvement strategies are the same. The second relationship, the equality for the best Pareto improvement, holds because all the solutions in  $\mathcal{N}$  are explored regardless of the initial Pareto set.  $\square$

### 3.2 Neutral Pareto improvement strategy, $\mathcal{I}_N$

The pseudo-code for the neutral Pareto improvement  $\mathcal{I}_N$  is presented in Algorithm 6.  $\mathcal{I}_N$  stops the exploration of a neighbourhood  $\mathcal{N}(s)$  when the first solution that is non-dominated by the initial solution  $s$  and non-dominated by the Pareto set  $A$  is found.

---

**Algorithm 6** Neutral Pareto Improvement  $\mathcal{I}_N(s, A, \mathcal{N}, \preceq, \mathbf{f})$

---

```

 $A' \leftarrow \{s\} \cup A$ 
for all  $s' \in \mathcal{N}(s)$  do
  if  $\forall s'' \in A', \mathbf{f}(s') \prec \mathbf{f}(s'') \vee \mathbf{f}(s') \parallel \mathbf{f}(s'')$  then
     $s'.visited \leftarrow false$ 
     $A' \leftarrow \text{merge}(A', \{s'\})$ 
     $A' \leftarrow A' \setminus (\{s\} \cup A)$ 
  Return  $A'$ 
end if
end for
return  $\emptyset$ 

```

---

**Definition 3** Let  $\mathcal{I}_N(s, A)$  be the neutral Pareto improvement strategy from Algorithm 6 and  $\pi_{\mathcal{N}(s)}$  denote a random permutation of  $\mathcal{N}(s)$ .  $\mathcal{I}_N$  stops after the first solution  $a \in \pi_{\mathcal{N}}$  is found such that  $a$  is incomparable with  $s$  and incomparable with all solutions from  $A$ . If  $a$  exists, the outputted Pareto set is  $A' \leftarrow \{a\}$ , and  $a.visited \leftarrow false$ . Otherwise,  $A' \leftarrow \emptyset$ .

Again, the initial solutions  $\{s\} \cup A$  are not contained in the output  $A'$  of  $\mathcal{I}_N$ . In the following proposition, we show that, even after merging the output of  $\mathcal{I}_N$  with  $A$ ,  $\mathcal{I}_N(s, A)$  is at least as good—meaning equal or dominating—with  $\mathcal{I}_N(s, \emptyset)$ .

**Proposition 3** Let  $\mathcal{I}_N(s, \emptyset)$  be the neutral Pareto improvement strategy from Definition 3. Then, the outputted Pareto sets of  $\mathcal{I}_N(s, A)$  and  $\mathcal{I}_N(s, \emptyset)$  are in the following relationships

1. If  $\mathcal{I}_N(s, A) = \{a\}$ , then  $\mathcal{I}_N(s, A) \prec \mathcal{I}_N(s, \emptyset)$  or  $\mathcal{I}_N(s, A) = \mathcal{I}_N(s, \emptyset)$ . Otherwise, if  $\mathcal{I}_N(s, A) = \emptyset$  and  $\mathcal{I}_N(s, \emptyset) = \{s'\}$ ,  $\exists s'' \in A$ , such that  $\mathbf{f}(s'') \prec \mathbf{f}(s')$
2.  $\text{merge}(\mathcal{I}_N(s, A), A) \preceq \text{merge}(\mathcal{I}_N(s, \emptyset), A)$

*Proof* Let  $s'$  and  $s'' \in \pi_{\mathcal{N}}$  be two solutions such that: (i)  $s'$  is the first and  $s''$  is the second solution in  $\pi_{\mathcal{N}(s)}$  that are incomparable with  $s$ , and (ii)  $s'$  is dominated by at least a solution in  $A$ , and (iii)  $s''$  is incomparable or dominating all solutions in  $A$ . Then,  $\mathcal{I}_N(s, \emptyset)$  stops at  $s'$  and  $\mathcal{I}_N(s, A)$  selects  $s''$ . For the rest of the cases, the output of  $\mathcal{I}_N(s, A)$  and  $\mathcal{I}_N(s, \emptyset)$  are equal. We conclude that the output of  $\mathcal{I}_N(s, A)$  is non-dominated by the output of  $\mathcal{I}_N(s, \emptyset)$ . Otherwise, the output of  $\mathcal{I}_N(s, \emptyset)$  is dominated by  $A$ . □

This means that there are cases where neutral Pareto improvement is outperforming first non-dominated solution (Liefoghe et al. 2011) but there are no situations where  $\mathcal{I}_N$  is outperformed by the first non-dominated solution.

### 3.3 First Pareto improvement strategy, $\mathcal{I}_F$

The pseudo-code for  $\mathcal{I}_F$  is given in Algorithm 7. In  $\mathcal{I}_F$ , the exploration of the neighbourhood stops when the first solution that dominates the initial solution is found.

---

**Algorithm 7** First Pareto Improvement  $\mathcal{I}_F(s, A, \mathcal{N}, \preceq, \mathbf{f})$

---

```

A' ← {s} ∪ A
for all s' ∈ N(s) do
  if ∀s'' ∈ A', f(s') < f(s'') ∨ f(s') || f(s'') then
    s'.visited ← false
    A' ← merge(A', {s'})
    if f(s') < f(s) then
      A' ← A' \ ({s} ∪ A)
    Return A'
  end if
end if
end for
A' ← A' \ ({s} ∪ A)
Return A'

```

---

**Definition 4** Let  $\mathcal{I}_F(s, A)$  be the *first Pareto improvement* strategy from Algorithm 7 and let  $\pi_{\mathcal{N}(s)}$  denote a random permutation of  $\mathcal{N}(s)$ .  $\mathcal{I}_F$  stops at the first solution  $a \in \mathcal{N}(s)$  that dominates  $s$ . If  $\exists a$ , the output  $A'$  contains  $a$  and all the solutions from  $\pi_{\mathcal{N}(s)}$  that precede  $a$  and that are: (i) incomparable with  $a$  and (ii) dominating or are incomparable with all solutions from  $\{s\} \cup A$  and all the other solutions from  $\pi_{\mathcal{N}(s)}$  that precede  $a$ . Otherwise, if  $\neg \exists a$ , then  $A'$  contains all the solutions from  $\mathcal{N}(s)$  that are: (i) incomparable with  $s$ , (ii) incomparable or dominate all solutions in  $A$ , and (iii) incomparable or dominating all other solutions in  $\mathcal{N}(s)$ . The visited flag of all solutions included in  $A'$  is set on *false*.

Like for the other two improvement strategies, from the output of  $\mathcal{I}_F$  are deleted the initial set of solutions  $\{s\} \cup A$ . Similarly with  $\mathcal{I}_B$ , the Pareto set of  $\mathcal{I}_F(s, A)$  is smaller or equal with the output of  $\mathcal{I}_F(s, \emptyset)$ , where the extra solutions are dominated by  $A$ . After merging the outputs of  $\mathcal{I}_F$  with  $A$ , the resulting Pareto sets are equal.

**Proposition 4** Let  $\mathcal{I}_F$  be as in Definition 4. Then, we have

1.  $\mathcal{I}_F(s, A) \subseteq \mathcal{I}_F(s, \emptyset)$ , and  $\forall s' \in \mathcal{I}_F(s, \emptyset) \setminus \mathcal{I}_F(s, A)$ ,  $\exists s'' \in A$ , such that  $\mathbf{f}(s'') \prec \mathbf{f}(s')$
2.  $\text{merge}(\mathcal{I}_F(s, \emptyset), A) = \text{merge}(\mathcal{I}_F(s, A), A)$

*Proof* Let us assume that  $\exists a \in \mathcal{N}(s)$  such that  $\mathbf{f}(a) \prec \mathbf{f}(s)$ . Then,  $a$  is non-dominated by the currently explored solutions for  $\pi_{\mathcal{N}(s)}$  and all solutions from  $A$ . Compared with  $\mathcal{I}_F(s, A)$ ,  $\mathcal{I}_F(s, \emptyset)$  returns solutions that are incomparable with  $a$  but dominated by a solution from  $A$ . Assume that  $\neg \exists a \in \mathcal{N}(s)$ ,  $\mathbf{f}(a) \prec \mathbf{f}(s)$ , then *all* the solutions in  $\mathcal{N}(s)$  are evaluated regardless of the initial Pareto set. Again, both strategies return the same result. □

The first improvement strategy (Liefvooghe et al. 2011) is slightly different from  $\mathcal{I}_F(s, \emptyset)$ . With the first improvement *all* non-dominated solutions with  $s$  are proposed

---

**Algorithm 8** Hyper-base first improvement  $I_H(s, A, \mathcal{N}, \geq, \mathcal{H})$

---

```

 $A' \leftarrow \{s\} \cup A$ 
for all  $s' \in \mathcal{N}(s)$  do
  if  $\mathcal{H}(r; A') < \mathcal{H}(r; \text{merge}(\{s'\}, A'))$  then
     $s'.visited \leftarrow \text{false}$ 
     $A' \leftarrow \text{merge}(A', \{s'\})$ 
     $A' \leftarrow A' \setminus (\{s\} \cup A)$ 
  Return  $A'$ 
end if
end for
return  $\emptyset$ 

```

---

for integration with  $A$  that includes solutions dominated by  $a$  or by solutions in  $A$ . With  $\mathcal{I}_F(s, \emptyset)$ , the solutions that are dominated by  $a$  are already deleted from the output. This makes a difference in computational time for large neighbourhoods with lots of incomparable solutions.

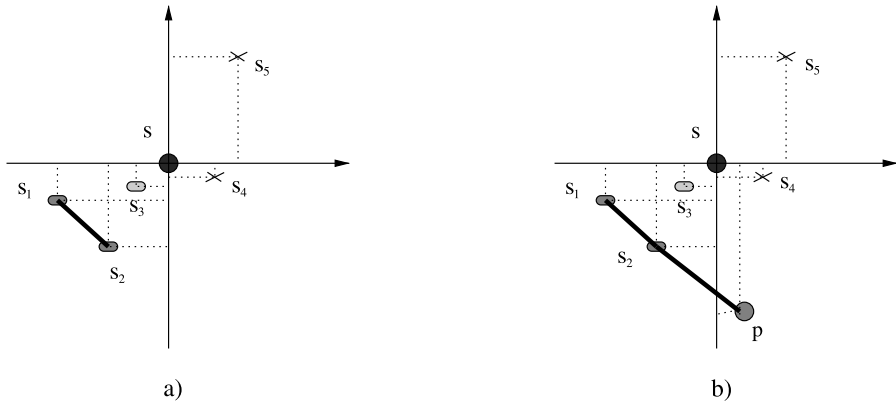
### 3.4 Pareto improvement strategies vs searching with the hypervolume unary indicator

The use of the initial Pareto set  $A$  allows us to compare the Pareto improvement strategies with a single-objective first improvement exploration strategy using the hypervolume unary indicator, a technique currently often used for multi-objective optimization (Zitzler et al. 2003; Bringmann and Friedrich 2010). This observation is important because some popular algorithms use the hypervolume indicator. For example, Emmerich et al. (2005), Beume et al. (2007), Bader and Zitzler (2011) use the hypervolume to select a subset of solutions from a larger set or to discard a solution from a set.

The hypervolume unary indicator function,  $\mathcal{H}$ , measures in the objective space the hypervolume contained between a reference point,  $r$ , and the Pareto set  $A$ . The hypervolume  $\mathcal{H}(r; A)$  is maximized, meaning that the larger the hypervolume is, the “better” its set  $A$  is considered. For a given reference point the hypervolume indicator of  $A$  is by design equal or smaller than the hypervolume indicator of  $(\{s\} \cup A)$  (Bader 2010). Algorithm 8 gives the pseudo-code for the hypervolume-based first improvement strategy.

**Definition 5** Let the hypervolume  $\mathcal{H}(r; A)$  act on the reference point  $r$  and a Pareto set  $A$ . The *hypervolume-based first improvement strategy* from Algorithm 8,  $\mathcal{I}_H(s, A, \mathcal{N}, \geq, \mathcal{H})$ , evaluates the hypervolume indicators of each of the sets  $\text{merge}(\{s'\}, \{s\} \cup A)$ , where  $s' \in \pi_{\mathcal{N}}$ , a random permutation of  $\mathcal{N}(s)$ .  $\mathcal{I}_H$  stops after the *first* solution  $a \in \pi_{\mathcal{N}}$  for which  $\mathcal{H}(r; A) < \mathcal{H}(r; \text{merge}(\{a\}, A))$ . If  $\exists a$ , the output is  $A' \leftarrow \{a\}$ . Otherwise,  $A' \leftarrow \emptyset$ .

We show the equivalence of  $\mathcal{I}_N$  with  $\mathcal{I}_H$  for any reference point  $r$ .



**Fig. 2** Consider  $\mathcal{N}(s) = \{s_1, s_2, s_3, s_4, s_5\}$ . We compare the three Pareto neighbourhood exploration strategies (a) with an empty initial Pareto set or (b) with the initial Pareto set  $A = \{p\}$

**Proposition 5** Let  $\mathcal{I}_N$  be the neutral Pareto improvement strategy from Definition 3 and let  $\mathcal{I}_H$  be the hypervolume-based first improvement strategy from Definition 5.

Then,  $\mathcal{I}_N(s, A, \mathcal{N}, \leq, \mathbf{f}) = \mathcal{I}_H(s, A, \mathcal{N}, \geq, \mathcal{H})$ .

*Proof* We split the proof in two parts. First, we show that  $\mathcal{I}_N \subseteq \mathcal{I}_H$ , and then we show that  $\mathcal{I}_H \subseteq \mathcal{I}_N$ .  $\mathcal{I}_N$  stops at the first generated solution  $s_1 \in \mathcal{N}(s)$  for which  $\mathbf{f}(s_1) \neq \mathbf{f}(s)$ , and  $\neg \exists s' \in A, \mathbf{f}(s') < \mathbf{f}(s_1)$ . The new Pareto set is  $\text{merge}(\{s_1\}, \{s\} \cup A)$  for which the hypervolume indicator of  $\mathcal{H}(r; \text{merge}(\{s_1\}, \{s\} \cap A))$  is larger than  $\mathcal{H}(r; \{s\} \cap A)$ . Then,  $\mathcal{I}_H$  selects  $s_1$  and stops.  $\mathcal{I}_H$  stops at the first generated solution  $s_2 \in \mathcal{N}(s)$  such that  $\mathcal{H}(r; \{s\} \cup A) < \mathcal{H}(r; \text{merge}(\{s_2\}, \{s\} \cup A))$ . This means that  $\{s_2\}$  is either: (i) incomparable with  $s$  and all other points in  $A$ , (ii)  $s_2$  dominates  $s$ . In both cases,  $\mathcal{I}_N$  selects  $s_2$ . We conclude that  $\mathcal{I}_N(s, \mathcal{N}, A, \leq, \mathbf{f}) = \mathcal{I}_H(s, A, \mathcal{N}, \geq, \mathcal{H})$ .  $\square$

To summarize, the output of local search that uses the hypervolume-based first improvement indicator is equivalent with the output of PLS with the neutral Pareto improvement strategy, but the computational cost of using the hypervolume indicator is much higher than the cost of running PLS with the neutral Pareto neighbourhood exploration.

### 3.5 Comparing Pareto neighbourhood exploration strategies

The behaviour of the three Pareto neighbourhood exploration strategies— $\mathcal{I}_B, \mathcal{I}_F$ , and  $\mathcal{I}_N$ —are compared by means of an example.

*Example 2* In Fig. 2, we denote the solutions with points and crosses and Pareto sets with line segments. Consider five solutions in  $\mathcal{N}(s)$ : (i)  $s_1$  and  $s_2$  are dominating  $s$  and are incomparable with each other, (ii)  $s_3$  dominates  $s$  but is dominated by  $s_1$  and  $s_2$ , (iii)  $s_4$  is incomparable with  $s$  but dominated by  $s_1, s_2$  and  $s_3$ , (iv)  $s_5$  is dominated by all other solutions. In Fig. 2(b), additionally to the solutions from Fig. 2(a), the initial Pareto set is  $A = \{p\}$ , where  $p$  is incomparable with both  $s_1$  and  $s_2$ .

Both  $\mathcal{I}_B(s, \emptyset)$  in Fig. 2(a) and  $\mathcal{I}_B(s, \{p\})$  in Fig. 2(b) return the dominating solutions  $s_1$  and  $s_2$ ,  $A' = \{s_1, s_2\}$ .  $\mathcal{I}_F(s, \emptyset)$  and  $\mathcal{I}_F(s, \{p\})$  select one of the solutions that dominates  $s$ :  $s_1$ ,  $s_2$  or  $s_3$ . Then, the output is  $A' = \{s_1\}$ , or  $A' = \{s_2\}$ , or  $A' = \{s_3\}$ .  $\mathcal{I}_N(s, \emptyset)$  selects one of the four solutions that dominates or is incomparable with  $s$ :  $s_1$ ,  $s_2$ ,  $s_3$ , and  $s_4$ .  $\mathcal{I}_N(s, \{p\})$  selects one of the three solutions that dominates or is incomparable with  $\{s, p\}$ :  $s_1$ ,  $s_2$ , and  $s_3$ . In Fig. 2(b),  $s_4$  is not selected because it is dominated by solutions in  $A$ . Then,  $\mathcal{I}_N(s, \emptyset)$  is dominated by  $\mathcal{I}_N(s, \{p\})$ .  $s_5$  is never selected.

To study the differences between the three Pareto neighbourhood exploration strategies  $\mathcal{I}_B$ ,  $\mathcal{I}_N$  and  $\mathcal{I}_F$ , we identify the conditions under which these strategies return the same result. The following statements result directly from their definitions. Let  $\pi_{\mathcal{N}}$  be a permutation of  $\mathcal{N}(s)$  and let  $|\mathcal{I}(s, A)|$  be the number of solutions evaluated in the algorithm  $\mathcal{I}(s, A)$ . The output of  $\mathcal{I}_B(s, A)$  and  $\mathcal{I}_F(s, A)$  are the same Pareto set if and only if one of the following conditions is met:

1. there exists *no* solution  $s' \in \mathcal{N}(s)$  such that  $s'$  dominates  $s$ ,
2. there exists *exactly* one solution  $s' \in \mathcal{N}(s)$  that: (i) dominates  $s$ , (ii) dominates all solutions in  $\mathcal{N}(s) \setminus \{s'\}$ , and (iii) precedes in  $\pi_{\mathcal{N}(s)}$ , if exists, the other solutions that dominate  $s$  but are dominated by  $s'$ .

The output of  $\mathcal{I}_N(s, A)$  is the same as the output of  $\mathcal{I}_F(s, A)$  if and only if one of the following conditions hold:

3. there are in  $\mathcal{N}(s)$  only solutions that are dominated by at least a solution in  $\{s\} \cup A$ ,
4. there exists *exactly* one solution  $s' \in \mathcal{N}(s)$  that: (i) is incomparable or dominating all solutions in  $\{s\} \cap A$ , and (ii) all other solutions in  $\mathcal{N}(s) \setminus \{s'\}$  are dominated by at least a solution in  $\{s\} \cap \{A\}$

The output of  $\mathcal{I}_N(s, A)$  is the same as the output of  $\mathcal{I}_B(s, A)$  if and only if the above conditions 3 and 4 hold. It is interesting to note that the conditions under which the outputs of  $\mathcal{I}_F$  and  $\mathcal{I}_B$  are equal include the conditions for which the outputs of  $\mathcal{I}_F$  and  $\mathcal{I}_N$  are equal, which, at their turn are equal with the conditions for  $\mathcal{I}_B$  and  $\mathcal{I}_N$  to be equal.

In the following, we compare the number and the quality of the solutions returned when the three Pareto neighbourhood exploration strategies are used.

**Proposition 6** *Let  $\mathcal{I}_B$ ,  $\mathcal{I}_N$  and  $\mathcal{I}_F$  be Pareto neighbourhood exploration strategies, and  $\pi_{\mathcal{N}(s)}$  the same permutation of the evaluated neighbourhood as before. Let us consider  $|\mathcal{I}(s, A)|$  the number of solutions evaluated with  $\mathcal{I}$ . Then,*

$$|\mathcal{I}_N(s, A)| \leq |\mathcal{I}_F(s, A)| \leq |\mathcal{I}_B(s, A)| \quad \text{and} \quad \mathcal{I}_B(s, A) \preceq \mathcal{I}_F(s, A) \preceq \mathcal{I}_N(s, A)$$

*Proof*  $\mathcal{I}_B$  generates *all* the neighbours of  $s$  thus it generates *always* at least as many individuals as the other two exploration strategies,  $\mathcal{I}_F$  and  $\mathcal{I}_N$ . This also means that  $\mathcal{I}_B$  always finds the solution, if it exists, that dominates  $\{s\} \cap A$  and all the other solutions  $\mathcal{N}(s)$ . Thus  $\mathcal{I}_B(s, A) \preceq \mathcal{I}_F(s, A)$  and  $|\mathcal{I}_F(s, A)| \leq |\mathcal{I}_B(s, A)|$ . We have  $\mathcal{I}_F(s, A) \preceq \mathcal{I}_N(s, A)$  because  $\mathcal{I}_F$  stops in a dominating solution for  $s$ , if it exists, whereas  $\mathcal{I}_N$  stops in any non-dominated solution for  $s$  and  $A$ . For a permutation

$\pi_{\mathcal{N}(s)}$ ,  $|\mathcal{I}_N(s, A)| \leq |\mathcal{I}_F(s, A)|$  because  $\mathcal{I}_N$  is always stopping earlier than  $\mathcal{I}_F$  unless one of the conditions 3 or 4 hold in which case  $|\mathcal{I}_N(s, A)| = |\mathcal{I}_F(s, A)|$ .  $\square$

According to the above proposition, there is a trade-off between the number of evaluated solutions and the quality of the newly added solutions. The advantage of  $\mathcal{I}_F$  and  $\mathcal{I}_N$  is that they evaluate only a part of the neighbourhood, whereas  $\mathcal{I}_B$  evaluates the entire neighbourhood. The disadvantage of  $\mathcal{I}_F$  and  $\mathcal{I}_N$  is that the solutions selected, in general, are dominated by the solutions selected with  $\mathcal{I}_B$ . The solution(s) generated with  $\mathcal{I}_F$  are non-dominated by the solutions selected with  $\mathcal{I}_N$ . Note that the time complexity of evaluating a solution with any of the three Pareto neighbourhood exploration strategies presented before is linear with the number of objectives,  $m$ , and the number of solutions in the neighbourhood of a solution  $s$ ,  $\mathcal{N}(s)$ , and the number of solutions in the initial Pareto set  $A$ .

### 3.6 Soundness and completeness of PLS using Pareto improvement strategies

Paquete et al. (2007) show that their PLS is always stopping and, it returns a maximal Pareto local optimal set. Our version of PLS from Algorithm 1 differs in certain aspects from Paquete's PLS: (i) the multiple Pareto neighbourhood exploration strategies,  $\mathcal{I}$ , and (ii) the starting Pareto set  $A$ . In Paquete's PLS, the Pareto set  $A$  is reduced to a single, randomly generated solution and there is a single improvement strategy, the best (Pareto) improvement strategy. We show that PLS using one of the three Pareto improvement strategies and an initial Pareto set has the local optimality property—that is, it always stops in a Pareto local optimal set. For simplicity of the proof, we assume that the initial Pareto set  $A$  contains only solution,  $s$ , with the visited flag set on *true*. This proof follows the proof of Theorems 5.3 and 5.4 from (Paquete et al. 2007), but with different improvement strategies for PLS.

**Theorem 1** *In Algorithm 1, PLS( $A, \mathcal{I}_B$ ) and PLS( $A, \mathcal{I}_F$ ) always stop in a maximal Pareto local optimal set. PLS( $A, \mathcal{I}_N$ ) stops in a subset of a maximal Pareto local optimal set.*

*Proof* In the first part of this proof, we show that no matter in which sequence solutions are visited, the output contains always a set of solutions that dominates the other solutions. Let us assume that there are two solutions,  $s'$  and  $s''$  in  $A'$ , such that  $\mathbf{f}(s') < \mathbf{f}(s'')$  and both solutions are evaluated with the same PLS. If  $s''$  is visited first, it is added to  $A'$ . When  $s'$  is visited,  $s''$  is discarded from  $A'$ . If  $s'$  is visited first, it is added to  $A'$ . When  $s''$  is visited, it is not added to the archive because  $s''$  is dominated by  $s'$ . We conclude that  $s'$  is always added to  $A'$  regardless of the succession in which solutions are visited.

Second, we show that PLS using  $\mathcal{I}_B$  and  $\mathcal{I}_F$  are always stopping in a maximal Pareto local optimal set. Thus, there are no solutions in the neighbourhoods of the solutions from the Pareto local optimal set that are incomparable with all the solutions in this set and are not added to the Pareto set. This property is already proven by Paquete et al. (2007) for best improvement strategies where all the neighbourhoods from all the solutions in  $A'$  are explored entirely. The property also holds for  $\mathcal{I}_F$



because, like for  $\mathcal{I}_B$ , all the incomparable solutions from the neighbourhood of a solution in the Pareto local optimal set are visited.

Consider  $\mathcal{I}_N$  and a solution  $s \in A'$ . Consider there are two solutions  $s'$  and  $s''$  in  $\mathcal{N}(s)$  that are incomparable with  $s$  and all other solutions in  $A'$  but that are not in the neighbourhood of each other. If  $s'$  is visited before  $s''$ , then  $s'$  is added to  $A'$  but  $s''$  is discarded and  $s$  visited set to true and thus  $s''$  is not visited again. We conclude that PLS using  $\mathcal{I}_N$  stops in a subset of a maximal Pareto local optimum set.

Third, we show that PLS is always terminating. We show that a solution  $s$  eliminated from the current archive  $A'$  because it is dominated by a newly added solution  $s'$ , where  $\mathbf{f}(s') < \mathbf{f}(s)$ , will not be added a second time into the archive  $A'$ . Suppose that the same solution  $s$  is compared with solutions in  $A'$  a second time.  $A'$  contains solutions that are dominating or incomparable with  $s'$ . Thus,  $A'$  contains solutions that are dominating  $s$ , and  $s$  is not included in this archive. If there are no other solutions that dominate the solutions from  $A'$ , the PLS algorithm from Algorithm 1 stops by design.  $\square$

Note that all three strategies add the dominating solutions, keep the incomparable solutions and delete the dominated solutions from the current Pareto set  $A'$ . The difference between these three Pareto neighbourhood exploration strategies is their stopping criterion.  $\mathcal{I}_N$  stops the earliest, since it only needs to find a new incomparable solution.  $\mathcal{I}_B$  stops after all solutions are visited, and  $\mathcal{I}_F$  stops somewhere in-between  $\mathcal{I}_N$  and  $\mathcal{I}_B$ .

#### 4 Genetic PLS (GPLS) for multi-objective quadratic assignment problems (mQAPs)

In order to design the neighbourhood,  $\mathcal{N}$ , the stopping criterion,  $\mathcal{T}$ , and genetic operators for GPLS, we need to specify a type of multi-objective combinatorial optimization problem. In this section, we design instances of GPLS for mQAPs. The neighbourhood and the stopping criterion are applicable for all kinds of permutation problems. For an optimal performance, the genetic operators are designed for each problem they are applied on.

*Multi-objective QAPs* Single and multi-objective QAPs are NP-hard combinatorial optimization problems that model many real-world problems (i.e., scheduling, vehicle routing, etc.). Intuitively, QAPs can be described as the (optimal) assignment of  $n$  facilities to  $n$  locations. A distance is specified between each pair of locations, and for each pair of facilities the amount of materials (or flows) transported between these facilities is given. The goal is to find the assignment of facilities to locations that minimizes the sum of the products between distances and flows.

We consider multi-objective QAPs introduced by Knowles and Corne (2003). These mQAPs have for each objective different flow matrices and a single distance matrix. The flow matrices are correlated with some correlation  $\rho$ . Let us consider  $n$  facilities, a set  $\Pi(n)$  of all permutations of  $\{1, 2, \dots, n\}$  and the  $n \times n$  distance matrix  $D = (d_{ij})$ , where  $d_{ij}$  is the distance between location  $i$  and location  $j$ . We assume an  $m$  objective space, and  $m$  flow matrices  $B^k = (b_{ij}^k)$ , each with

$n \times n$  elements, where  $b_{ij}^k$  represents the flow from facility  $i$  to facility  $j$  in the  $k$ -th objective. The goal is to minimize for all objectives the set of *cost functions*  $c_k(\pi) = \sum_{i=1}^n \sum_{j=1}^n d_{ij} \cdot b_{\pi(i)\pi(j)}^k$ , where  $\pi(\cdot)$  is a permutation from  $\Pi(n)$ . It takes quadratic time to evaluate this function.

*The neighbourhood  $\mathcal{N}(\cdot)$*  Multi-objective QAPs are permutation problems. A suitable neighbourhood operator for mQAPs is the  $q$ -exchange operator that swaps the position of  $q$  facilities. For example, the 2-exchange swapping operator swaps the position of two different facilities. This operator is attractive because of its linear time to compute the change in the cost function with the condition that all matrices  $D$  and  $B^k$  are symmetrical (Paquete and Stützle 2006). We have  $\Delta c_k(\pi, i, j) = 2 \cdot \sum_{r=1, r \neq i, j}^n (d_{jk} - d_{ik}) \cdot (b_{\pi(i)\pi(k)}^k - b_{\pi(j)\pi(k)}^k)$ . This neighbourhood contains  $C_2^n = \binom{n}{2}$  individuals. The size of the neighbourhood increases quadratically with the number of facilities.

*Stopping criterion,  $\mathcal{T}$*  The stopping criterion for GPLS is chosen to fairly compare its performance with MPLS. The distance between two solutions is defined as the minimum number of exchanges needed to obtain one solution from the other solution (Schiavinotto and Stützle 2007). The distance between a solution and the solution obtained with  $q$ -exchange mutation is  $q - 1$ . The search in GPLS is halted when the same number of swaps is executed as with MPLS. The difference in cost function,  $\Delta c_k$ , for two solutions with distance  $q$  is linear in the number of facilities and the number of exchanges. Counting the number of swaps is equivalent to counting the number of fitness evaluations.

#### 4.1 Genetic operators for GPLS on mQAPs

In this section, we present two perturbation operators for mQAPs (Drugan and Thierens 2010). In Algorithm 4, the *Mutation* algorithm is the parametrized mutation, and the path-guided mutation has the role of *Recombination*. We use cycles (Schiavinotto and Stützle 2007) to design operators that generate children at a given distance from their parent. These types of operators allow us to analyse the behaviour of GPLS on the tested problem instances.

*The parametrized mutation, or the  $q$ -exchange mutation,* uniform randomly selects, without replacement,  $q > 2$  distinct locations in a solution  $s$ ,  $\{l_1, \dots, l_q\}$ . To generate a new solution, these locations are exchanged from left to right or from right to left with equal probability to not bias the generation of individuals. When positions are exchanged from right to left, a position  $l_i$  takes the value of its right neighbour  $l_{i+1}$ . Then, in this sequence,  $b \leftarrow l_1, l_1 \leftarrow l_2$  and so on until  $l_{q-1} \leftarrow l_q$  and  $l_q \leftarrow b$ , where  $b$  is a buffer variable. Note that  $s$  and the resulting solution form a cycle of size  $q$  that contains the mutated positions  $\{l_1, \dots, l_q\}$  and are  $q$  swaps apart. When PLS uses the 2-exchange operator to generate a neighbourhood, the mutation operator should exchange at least 3 facilities to escape from the region of attraction of a Pareto local optimal set.

The *path-guided mutation*, or the *q-exchange path-mutation*, uses two solutions,  $s'$  and  $s''$ , uniformly randomly selected without replacement from the current Pareto set such that the distance between them is at least  $q$ . A child  $s$  is generated by copying the first parent  $s'$ , and mutate  $s$ . The second parent  $s''$ , is used to construct the path between the two solutions. The set of *cycles* common for the two solutions,  $s$  and  $s''$ , are identified. A cycle is a minimal subset of locations such that the set of their facilities is the same in both parents,  $s'$  and  $s''$ . It is possible to switch a subset from one parent to the other one while keeping a valid permutation.

A cycle,  $c$ , is randomly chosen. For  $q - 1$  times, choose at random a position  $i$  in the cycle  $c$  from solution  $s$ , where  $s[i] = s''[j]$  and  $i \neq j$ . Exchange the values of  $s[i]$  and  $s[j]$ . With this swap, the distance between  $s$  and its first parent,  $s'$ , is increased with 1 and the distance between  $s$  and the second parent,  $s''$ , is decreased with 1. If the size of  $c$  is smaller or equal than  $q$ , a second cycle is chosen. This process of randomly selecting a cycle and swap locations is repeated until the distance between  $s$  and  $s'$  is becoming  $q$ . Then distance between  $s$  and  $s''$  is  $\ell - q + 1$ . If there are no parent solutions at distance larger or equal with  $q$ , we generate a solution with parametrized mutation.

The path-guided mutation is respectful: if the two parents have the same facility on a position  $i$  then their child will also have the same facility on the  $i$ -th position. Consequently, to be able to generate *any* solution in the search space we need to alternate path-guided mutation and mutation operators. The path-guided mutation operator resembles the *path relinking operator* (Jaszkiewicz and Zielniewicz 2009) because it generates solutions on the path between two solutions.

Cycle recombination (Poon and Carter 1995) uses cycles to generate children, but, unlike path-guided mutation, it exchanges an entire block of cycles between parents. Partially mapped crossover (PMX) (Goldberg and Lingle 1985) is the most popular recombination operator for the (multi-objective) travelling salesman problem (TSP). Some positions are uniformly randomly chosen in the two parents and information is exchanged between them such that the generated children are valid permutations. Unlike path-guided mutation, PMX creates, or connects, cycles rather than respecting them. In Sect. 6, we show that operators that do not respect cycles function worse than the operators proposed here that do respect them.

#### 4.2 The expected number of solutions compared with a Pareto improvement strategy

In this section we derive a formula to compute the expected number of solutions generated with the three Pareto neighbourhood exploration strategies. We show that the number of solutions generated with  $\mathcal{I}_N$  and  $\mathcal{I}_F$  are in expectation much smaller than with  $\mathcal{I}_B$ . Let us consider a neighbourhood  $\mathcal{N}(s)$  generated with the 2-exchange operator that has:

- $z$ —the number of solutions that dominate  $s$ ,  $z = |\{a \in \mathcal{N}(s) \mid \mathbf{f}(a) \prec \mathbf{f}(s)\}|$ , where  $0 \leq z \leq C_2^n$ ,
- $v$ —the number of solutions that are incomparable with  $s$ ,  $v = |\{a \in \mathcal{N}(s) \mid \mathbf{f}(s) \parallel \mathbf{f}(a)\}|$ , where  $0 \leq v \leq C_2^n$ ;
- $C_2^n - z - v$ —the number of solutions that are dominated by  $s$ ,  $C_2^n - z - v = |\{a \in \mathcal{N}(s) \mid \mathbf{f}(s) \prec \mathbf{f}(a)\}|$ .

The *best Pareto improvement* strategy evaluates *all* the solutions in the neighbourhood. The number of solutions evaluated with  $\mathcal{I}_B$  is  $|\mathcal{I}_B| = C_2^n$ . The probability of evaluating a certain number of solutions with  $\mathcal{I}_F$  follows a hyper-geometrical distribution describing the number of successes in a finite sequence where samples are drawn without replacement.

**Lemma 1** *Let  $\mathcal{I}_F$  and  $\mathcal{N}(s)$  be as before. Let  $z$  be the number of solutions in  $\mathcal{N}(s)$  that dominate  $s$ . The probability to find, after exploring exactly  $i$  solutions from  $\mathcal{N}(s)$ , the first solution that dominates  $s$  is*

$$P_{\mathcal{I}_F(s)}(x = i) = \frac{z}{i} * \binom{C_2^n - z}{i - 1} / \binom{C_2^n}{i} \tag{1}$$

*Proof*  $\mathcal{I}_F$  stops at the first improvement in the sequence of solutions generated from  $\mathcal{N}(s)$ . In total there are  $z$  improvements in this sequence of  $C_2^n$  solutions. Let successes be associated with improvements. The number of improvements, and thus success, before  $\mathcal{I}_F$  stops is 1. Using the definition of the hyper-geometrical distribution, we assume that there are  $z$  successes and  $C_2^n - z$  failures. The probability of finding the first improvement after  $i$  trials from the  $z$  total improvements is the hyper-geometrical value divided by  $i$ ,  $\binom{z}{1} \cdot \binom{C_2^n - z}{i - 1} / \binom{C_2^n}{i} / i$ . The probability  $P_{\mathcal{I}_F(s)}(x = i)$  follows directly.  $\square$

For the *neutral Pareto improvement* strategy, the probability to discover in  $i$  steps the first solution that dominates or is incomparable with  $s$  is

$$P_{\mathcal{I}_N(s)}(x = (i)) = \frac{z + v}{i} * \binom{C_2^n - z - v}{i - 1} / \binom{C_2^n}{i} \tag{2}$$

**Proposition 7** *Let  $\mathcal{I}_B, \mathcal{I}_F, \mathcal{I}_N$  be as before. Let  $\mathcal{N}$  be the neighbourhood generated with 2-exchange operator, and let  $z$  be the number of solutions that dominate  $s$ ,  $v$  the number of solutions that are incomparable with  $s$ , and  $\pi_{\mathcal{N}(s)}$  a uniform randomly generated permutation of  $\mathcal{N}(s)$  as before.*

*The expected number of evaluated neighbours with  $\mathcal{I}_B$  is*

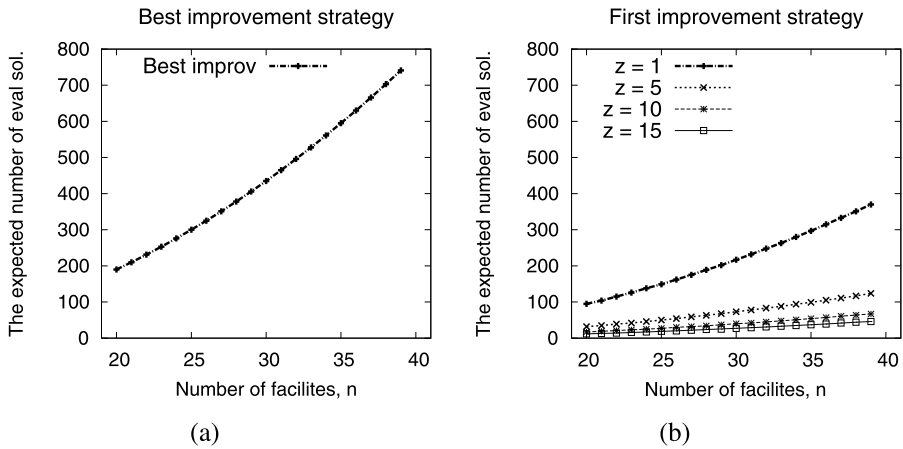
$$E\{|\mathcal{I}_B|\} = C_2^n$$

*where  $E\{|g(\cdot)|\}$  is the expected value of the function  $g$ . The expected number of evaluated neighbours with  $\mathcal{I}_F$  is*

$$E\{|\mathcal{I}_F|\} = z \cdot \sum_{i=1}^{C_2^n - z} \binom{C_2^n - z}{i - 1} / \binom{C_2^n}{i}$$

*The expected number of evaluated neighbours with  $\mathcal{I}_N$  is*

$$E\{|\mathcal{I}_N|\} = (z + v) \cdot \sum_{i=1}^{C_2^n - (z+v)} \binom{C_2^n - (z + v)}{i - 1} / \binom{C_2^n}{i}$$



**Fig. 3** The expected number of evaluated solutions in a neighborhood when (a)  $\mathcal{I}_B$  and (b)  $\mathcal{I}_F$  are used. The number of facilities  $n = \{20, \dots, 40\}$  and the number of dominating solutions  $z = \{1, 5, 10, 15\}$

*Proof* The number of evaluated neighbours is always the same for  $\mathcal{I}_B$ ,  $C_2^n$ . Thus,  $E\{|\mathcal{I}_B|\} = C_2^n$ . To calculate the expected evaluated solutions for  $\mathcal{I}_F$  and  $\mathcal{I}_N$ , we use the definition of the expected value for the function  $g(\cdot)$ ,  $E\{g\} = \sum_x g(x) \cdot P(x)$ . Here, the probability  $P(x = (i))$  is, as in Eq. 2, the probability that a Pareto improvement strategy generates  $i$  solutions before the first improvement on  $s$  is met. The sum is over all  $i = \{1, \dots, C_2^n - z\}$ , where  $C_2^n - z$  is the maximum number of solutions that can be evaluated before the first improvement on  $s$  is found. The function  $g(i) = i$  as usually. Then, the expected number of evaluated solutions for first Pareto improvement strategy is  $E\{|\mathcal{I}_F|\} = \sum_{i=1}^{C_2^n - z} g(i) \cdot P_{\mathcal{I}_F(s)}(x = (i))$ . To proof the expected value for  $\mathcal{I}_N$ , we follow the same line of reasoning.  $\square$

Some properties of these expected values are given in the following proposition.

**Proposition 8** *If  $n$  increases, then each of the expected values  $E\{|\mathcal{I}_N|\}$ ,  $E\{|\mathcal{I}_F|\}$ ,  $E\{|\mathcal{I}_B|\}$  also increases. If  $z$  increases, then  $E\{|\mathcal{I}_F|\}$  and  $E\{|\mathcal{I}_N|\}$  decrease. If  $v$  increases, then  $E\{|\mathcal{I}_N|\}$  decreases.*

*Proof* The first property follows directly from the definition of expected value. If  $z$  increases, then the terms  $C_2^n - z$ ,  $\binom{C_2^n - z}{i-1}$ ,  $C_2^n - (z + v)$  and  $\binom{C_2^n - (z+v)}{i-1}$ , from Eqs. 2 and 1, decrease. Thus,  $E\{|\mathcal{I}_F|\}$  and  $E\{|\mathcal{I}_N|\}$  decrease. Proving that when  $v$  increases,  $E\{|\mathcal{I}_N|\}$  decreases follows the same line of reasoning.  $\square$

Figure 3 shows the expected number of evaluated solutions in a neighbourhood with (a) best Pareto improvement and (b) first Pareto improvement for several values of the number of facilities,  $n$ , and the number of dominating solutions,  $z$ . The expected number of evaluated solutions in a neighbourhood is always largest for best Pareto improvement,  $E\{|\mathcal{I}_B|\}$ . Even if only one solution in the neighbourhood is dominating the current solution,  $z = 1$ ,  $E\{|\mathcal{I}_F|\}$  is half of  $E\{|\mathcal{I}_B|\}$ . As stated in

Proposition 8, the values of  $E\{\mathcal{I}_F\}$  decrease when  $z$  increases, but the expected values are increasing with  $n$ .

*“Flat” multi-objective landscapes* An interesting, but logical, property is that  $\mathcal{I}_N$  and  $\mathcal{I}_F$  return less solutions in a flat landscape (i.e. landscape with many incomparable solutions) than  $\mathcal{I}_B$ . Single objective spaces with many equal values are called *flat* and they are known as inappropriate for heuristics in general. Multi-objective spaces with many incomparable values are more common than flat single objective spaces. In general, all the subsets of elements  $\{x_i, \dots, x_j\} \in \mathcal{S}$  such that  $f_1(x_i) \leq \dots \leq f_1(x_j)$  and  $f_2(x_i) \geq \dots \geq f_2(x_j)$  form a Pareto set. For “flat” spaces, or spaces with a large “flat” region, when PLS uses  $\mathcal{I}_B$ , the entire region is explored and added to the current Pareto set. That leads to computational and storage problems for PLS. In this case, a Pareto improvement strategy, like  $\mathcal{I}_F$  and  $\mathcal{I}_N$ , that finds a single or few solutions from the space is less demanding in memory and run-time than  $\mathcal{I}_B$ .

## 5 Experimental results

In this section, we experimentally compare six different (stochastic) Pareto local search algorithms. The first three are multi-start PLS algorithms (MPLS) each using a different Pareto neighbourhood exploration strategy: best improvement  $\mathcal{I}_B$ , first improvement  $\mathcal{I}_F$ , and neutral improvement  $\mathcal{I}_N$ . The other three are SPLS algorithms that combine the above three neighbourhood exploration strategies with stochastic perturbation that consists of unbiased mutation and path-guided mutation, each applied with equal probability. All algorithms generate their neighbourhood  $\mathcal{N}$  with the 2-exchange operator. We denote the six algorithms as follows:

1. bMPLS uses  $\text{MPLS}(\mathcal{I}_B, \mathcal{T})$ , where  $\mathcal{T}$  means that PLS is restarted for  $M = 100$  times;
2. fMPLS uses  $\text{MPLS}(\mathcal{I}_F, \mathcal{T}')$ , where  $\mathcal{T}'$  means that PLS is restarted until the same number of swaps are generated as with bMPLS;
3. nMPLS uses  $\text{MPLS}(\mathcal{I}_N, \mathcal{T}')$ , with  $\mathcal{T}'$  as in fMPLS;
4. bGPLS uses  $\text{GPLS}(\mathcal{I}_B, 0.5, \mathcal{T}_1, \mathcal{T}_2)$ , where  $\mathcal{T}_1$  randomly restarts PLS  $N = 10$  times and  $\mathcal{T}_2$  means that PLS is restarted using genetic operators until the same number of swaps are generated as with bMPLS;
5. fGPLS uses  $\text{GPLS}(\mathcal{I}_F, 0.5, \mathcal{T}_1, \mathcal{T}_2)$ , with  $\mathcal{T}_1$  and  $\mathcal{T}_2$  as in bGPLS;
6. nGPLS uses  $\text{GPLS}(\mathcal{I}_N, 0.5, \mathcal{T}_1, \mathcal{T}_2)$ , with  $\mathcal{T}_1$  and  $\mathcal{T}_2$  as in bGPLS.

We generate restarting solutions with both mutation and path-guided mutation with an exchange rate  $q$ , uniform randomly selected between 3 and  $\lfloor n/3 \rfloor$ . These exchange rates are chosen based on the empirical observations from Drugan and Thierens (2010, 2011). Also the landscape analysis in the paper shows that these exchange rates are the most effective. For example, for  $n = 25$ , we have  $q = \{3, \dots, 8\}$ , and for  $n = 100$ , we have  $q = \{3, \dots, 33\}$ . Each algorithm independently runs 50 times.

*The tested problems* Experimental tests are performed on a set of bi-objective QAP instances (Knowles and Corne 2003) with positive correlations  $\rho = \{0.25, 0.50, 0.75\}$  and a large number of facilities  $n = \{25, 50, 75, 100\}$ . To facilitate comparisons, some of these problems are the unstructured bQAP instances in Paquete's study (Paquete and Stützle 2006) (<http://eden.dei.uc.pt/~paquete/qap/>). Three bQAPs, with  $n = 100$  and  $\rho = \{0.25, 0.5, 0.75\}$ , are generated with the software from Knowles and Corne (2003). In general, we denote the bQAP with  $\text{bQAP}(n, \rho)$ , with  $n$  and  $\rho$  as before. For QAPs with a large number of facilities and high positive correlation,  $\rho = 0.75$ , Paquete et al. reported poor performance of multi-restart PLS.

It should be noted that we have performed preliminary experiments with unstructured and structured with negative and zero correlation bQAPs. As Paquete et al. noticed, on negative unstructured bQAP instances more complicated LS-based heuristics and the multi-restart PLS have approximatively the same performance. The Pareto sets of these bQAPs are very large, for instance for  $\rho = 0$  and  $n = 75$  the Pareto optimal set typically found has about 2000 solutions. This can be explained by the fact that when there is no correlation, many solutions have a different ordering for the two objectives. Similarly when there is negative correlation ( $\rho = -0.75$ ), the sizes of the Pareto optimal sets typically found are huge. In this case, the ordering of each solution is opposite in the two objectives. For such large Pareto optimal sets the search with PLS using  $\mathcal{T}_B$  therefore becomes too expensive. Knowles and Corne (2004) acknowledge that the structured QAPs are easier than the unstructured bQAPs for local search.

To summarize, we consider that the complexity is highest when there are a large number of attractors containing Pareto optimal solutions, compared to few attractors with a large number of Pareto optimal solutions. Section 5.1 looks at the performance of the six algorithms, Sect. 5.2 discusses their exploration properties, while Sect. 5.3 focuses on their dynamical behaviour.

### 5.1 Comparing the performance of stochastic PLS algorithms

Pareto sets are typically compared by the hypervolume unary indicator and/or attainment functions (Zitzler et al. 2003). The unary *attainment function* (Fonseca and Fleming 1996) (EAF) gives the probability of attaining each point (independently) in the objective space. Contour surfaces through certain probabilities can then be drawn. To compute the hypervolume unary indicator, we use the PISA package (Bleuler et al. 2003). For comparison purposes we first normalize the outputs of all six algorithms, and the corresponding reference set, if it exists, from <http://eden.dei.uc.pt/~paquete/qap/>,  $A^*(n, \rho)$ , where  $n = \{0.25, 0.5, 0.75\}$  and  $\rho$  as before. For each instance, the reference set  $A^*$  is obtained with a two phase tabu algorithm that is run much longer than the MPLSs are run. There are algorithms that outperform this tabu algorithm in some cases (Lopez-Ibanez et al. 2004). This normalization script assigns to the smallest point(s) in an objective the value 1 and to the highest point(s) the value 2. All the other points are scaled to a value between 1 and 2 in both objectives. The normalized outputs are the inputs for the hypervolume calculation. The reference point is (2.1, 2.1) which has the largest value in both objectives.

**Table 2** The performance of the six SPLS algorithms on 12 bQAP instances

bQAP		(A) The unary hypervolume indicator					
<i>n</i>	$\rho$	bMPLS	bGPLS	fMPLS	fGPLS	nMPLS	nGPLS
25	0.25	<b>0.95 ± 0.02</b>	0.94 ± 0.01	<b>0.95 ± 0.01</b>	<b>0.95 ± 0.01</b>	0.89 ± 0.02	0.94 ± 0.02
25	0.5	0.92 ± 0.03	0.94 ± 0.03	0.94 ± 0.03	<b>0.97 ± 0.03</b>	0.87 ± 0.04	0.95 ± 0.03
25	0.75	0.71 ± 0.08	0.80 ± 0.11	0.77 ± 0.09	<b>0.88 ± 0.09</b>	0.75 ± 0.08	0.87 ± 0.10
50	0.25	0.88 ± 0.01	<b>0.93 ± 0.02</b>	0.88 ± 0.01	<b>0.93 ± 0.02</b>	0.83 ± 0.02	0.86 ± 0.02
50	0.5	0.79 ± 0.02	0.89 ± 0.04	0.80 ± 0.02	<b>0.91 ± 0.03</b>	0.71 ± 0.03	0.85 ± 0.04
50	0.75	0.53 ± 0.05	0.79 ± 0.10	0.57 ± 0.06	<b>0.88 ± 0.11</b>	0.54 ± 0.06	0.81 ± 0.09
75	0.25	0.87 ± 0.02	<b>0.94 ± 0.02</b>	0.86 ± 0.01	<b>0.94 ± 0.02</b>	0.74 ± 0.00	0.84 ± 0.02
75	0.5	0.75 ± 0.02	0.89 ± 0.03	0.76 ± 0.02	<b>0.92 ± 0.04</b>	0.64 ± 0.02	0.82 ± 0.04
75	0.75	0.42 ± 0.07	0.77 ± 0.10	0.45 ± 0.04	<b>0.87 ± 0.10</b>	0.40 ± 0.05	0.79 ± 0.12
100	0.25	0.89 ± 0.01	0.98 ± 0.02	0.89 ± 0.01	<b>0.99 ± 0.01</b>	0.85 ± 0.01	0.86 ± 0.03
100	0.5	0.74 ± 0.02	0.95 ± 0.03	0.76 ± 0.02	<b>0.97 ± 0.02</b>	0.66 ± 0.02	0.84 ± 0.05
100	0.75	0.42 ± 0.05	0.83 ± 0.11	0.44 ± 0.14	<b>0.89 ± 0.12</b>	0.41 ± 0.05	0.82 ± 0.11

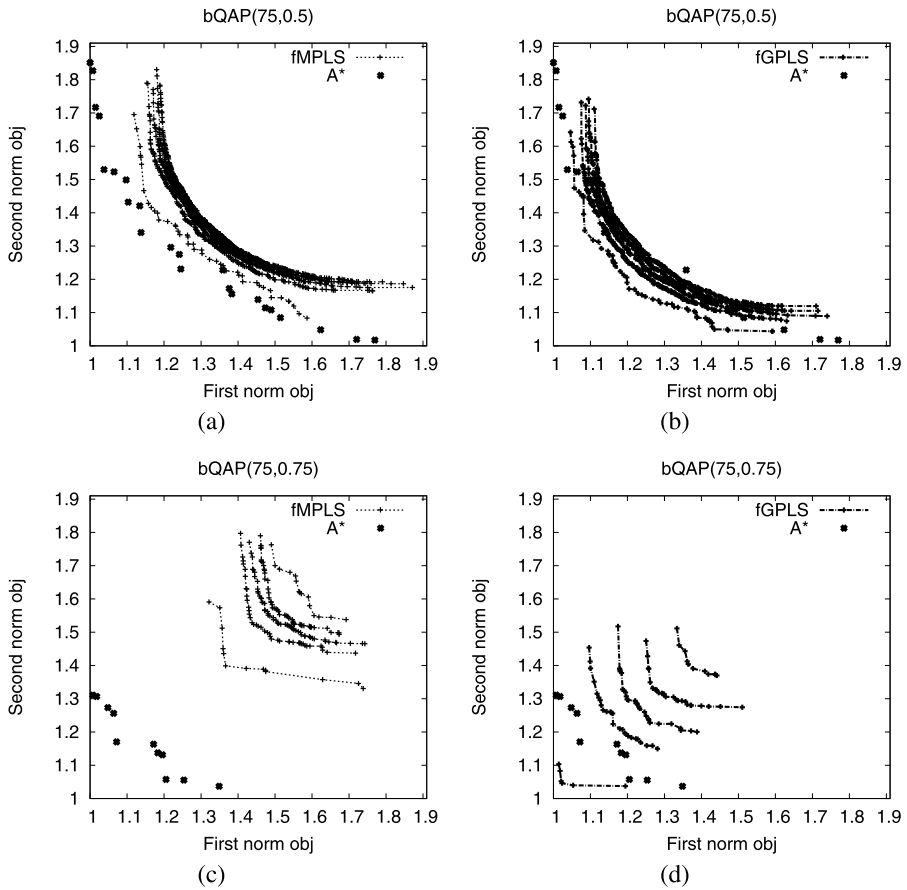
  

SPLS	(B) Compare the six SPLS algorithms										
	Mean							# wins			Rank
	$\rho$			<i>n</i>				SPLS			
	0.25	0.5	0.75	25	50	75	100	fGPLS/	bGPLS/	fMPLS/	
fGPLS	0.95	0.94	0.88	0.93	0.91	0.91	0.95	–	0	0	1
bGPLS	0.90	0.92	0.80	0.89	0.87	0.87	0.86	9	–	0	2
nGPLS	0.88	0.87	0.82	0.92	0.84	0.82	0.84	11	7	8	3
fMPLS	0.90	0.82	0.56	0.89	0.75	0.69	0.70	11	9	–	4
bMPLS	0.90	0.80	0.52	0.86	0.73	0.68	0.68	11	10	3	5
nMPLS	0.83	0.72	0.53	0.84	0.69	0.59	0.64	12	12	10	6

Table 2(A) shows the average and the standard deviation of the hypervolume unary indicators of the six SPLSs on twelve testing bQAPs. Table 2(B) compares the six SPLS algorithms: (i) the mean hypervolume for the three values of  $\rho$  and four numbers of facilities,  $n$ , (ii) the number of wins of fGPLS, bGPLS and fMPLS over the six SPLSs, and (iii) the rank of each SPLS algorithms based on the number of wins. For the same bQAP instance, we measure if the difference between two measurements of Table 2(A) is statistical significant using the Wilcoxon non-parametric two samples test, where  $p < 0.05$ . Figure 4 presents five EAFs at 2%, 25%, 50%, 75% and 100% for two algorithms, fMPLS and fGPLS, on two bQAPs, bQAP(75, 0.5) and bQAP(75, 0.75). Overall, fGPLS is the best tested SPLS and nMPLS is the worst algorithm.

All GPLS algorithms -that are fGPLS, nGPLS, bGPLS—outperform all the multi-restart PLS—that are bMPLS, fMPLS and nMPLS—on the twelve tested bQAPs. In Fig. 4, the EAFs contours are closer to the lowest point (1, 1), and thus better, for fGPLS than for fMPLS. Furthermore, for fGPLS the best EAF contour at 2% is





**Fig. 4** Attainment surfaces at 2 %, 25 %, 50 %, 75 % and 100 % (lines from bottom left to top right) of the normalized outputs. The two tested SPLSs are: (i) fMPLS on the left and (ii) fGPLS on the right. The two tested bQAPs are: (i) bQAP(75, 0.5) top and (ii) bQAP(75, 0.75) bottom. These EAFs are compared with the best known Pareto set for the two bQAPs: (i)  $A^*(75, 0.5)$  top and (ii)  $A^*(75, 0.75)$  bottom. The objective values found at 2 % EAFs correspond to the best Pareto set found over 50 runs and those found at 50 % EAFs are the median outcome

better than 1 % attainment surface for  $A^*$  from 100 runs on both bQAP instances. The objective values of 2 % EAFs are better for  $A^*$  than for fMPLS. Thus, the two perturbation operators of GPLS increase the efficiency of the SPLS algorithms.

Table 2(B) shows that the SPLS algorithms that use perturbation operators have the highest performance, rank 1, 2, and 3. The difference between GPLS and its corresponding MPLS with the same Pareto neighbourhood exploration strategy increases when the correlation  $\rho$  and the number of facilities  $n$  increase. For each SPLS, the hypervolume indicator decreases with the increase of  $\rho$  and  $n$ . The smallest, statistically the same, hypervolume values are registered with the three MPLSs on bQAP(100, 0.75). On the same bQAP instance, the smallest difference between the hypervolume of GPLS and MPLS is 0.41 when  $\mathcal{I}_B$  and  $\mathcal{I}_N$  are used. The largest hy-

pervolume difference, 0.45, is between fGPLS and fMPLS. Note that, on bQAP(100, 0.75) the hypervolume indicators of GPLSs are approximately two times larger than the hypervolume indicators of MPLSs. On the contrary, on bQAP(25, 0.25) all SPLS algorithms perform about the same, the exception is nMPLS, which has a lower performance. We conclude that the difference between the performance of GPLS and MPLS increases with increasing problem complexity—that is the increase in  $n$  and  $\rho$ .

In Fig. 4, on bQAP(75, 0.5), attainment surfaces, EAFs, are clustered together, whereas on bQAP(75, 0.75) they are spread. Attainment surfaces of fGPLS are more spread than the attainment functions of fMPLS. In Table 2, the clustered EAFs correspond to a small variance of hypervolume values whereas the spread EAFs correspond to large variances of the hypervolume indicators. On the other side, on bQAP(75, 0.5), in Fig. 4(top), each attainment surface is reasonably well spread over the two objectives. On bQAP(75, 0.75), in Fig. 4(bottom), the attainment surfaces are rather narrow and have a small amount of points. For large correlations, there are few incomparable solutions and a large difference between the worst and the best objective values. For small correlations, there is a large amount of solutions with different precedence in the two objectives and a smaller difference between the worst and the best objectives.

The SPLS algorithms that use  $\mathcal{I}_F$  have a better or equal performance than the corresponding SPLSs that use  $\mathcal{I}_B$ . In Table 2(B), both the best genetic PLS, rank 1, and the best multi-restart PLS, rank 4, use  $\mathcal{I}_F$ . The SPLSs that use  $\mathcal{I}_F$  or  $\mathcal{I}_B$  have a better or equal performance than the corresponding SPLSs that use  $\mathcal{I}_N$ . For bQAPs with low correlation,  $\rho = 0.25$ , there are no statistical differences between algorithms using  $\mathcal{I}_B$  and  $\mathcal{I}_F$ . bQAPs with high correlation,  $\rho = 0.75$ , give the largest difference between SPLSs that use  $\mathcal{I}_F$  and  $\mathcal{I}_B$ . For bQAPs with  $\rho = \{0.25, 0.5\}$ , SPLSs that use  $\mathcal{I}_N$  are the algorithms with the worst performance. The bad performance of SPLSs using  $\mathcal{I}_N$  is due to the long walk on the Pareto set of incomparable solutions instead of searching for dominating solutions that greatly increase the hypervolume. For  $\rho = 0.75$ , SPLSs that use  $\mathcal{I}_N$  have similar performance with SPLSs that use  $\mathcal{I}_B$ . The small number of solutions in the Pareto sets of bQAPs with  $\rho = 0.75$ , restricts the time  $\mathcal{I}_N$  spends in a neighbourhood of a solution.

The overall conclusion is that the SPLS algorithm with the highest performance is fGPLS using the first improvement Pareto neighbourhood exploration  $\mathcal{I}_F$  and both perturbation methods, mutation and path-guided mutation. The bQAP, where the difference between fGPLS and the other SPLSs is maximal, has the largest number of facilities and the largest correlation, bQAP(100, 0.75). In the next section, we give more insights on the SPLSs' behaviours on the tested bQAPs.

## 5.2 Landscape characteristics of bQAPs explored with SPLSs

Table 3 shows properties of bQAPs when explored with SPLSs. Table 3(A) shows the number of times PLS is restarted in a SPLS run, #PLS, where all the SPLS algorithms have the same run-time, measured in number of swaps. Table 3(B) shows that the number of calls of the Pareto improvement strategy, # $\mathcal{I}$  is proportional with  $n$  and inverse proportional with  $\rho$ . In Table 3(C), the number of solutions in the output Pareto local optimal set of a PLS,  $|A|$ , is proportional with  $n$  and inverse proportional with  $\rho$ .

**Table 3** Characterizing the bQAP's landscape

$n$	$\rho$	bMPLS	bGPLS	fMPLS	fGPLS	nMPLS	nGPLS	Mean (rk)
bQAP		(A) The number of times PLS is restarted, #PLS						
25	0.25	100 ± 0	326 ± 26	107 ± 3	321 ± 20	682 ± 15	1397 ± 46	484 (6)
25	0.5	100 ± 0	335 ± 49	116 ± 4	328 ± 39	392 ± 9	672 ± 32	324 (8)
25	0.75	100 ± 0	334 ± 49	157 ± 4	379 ± 52	230 ± 5	441 ± 43	274 (12)
50	0.25	100 ± 0	342 ± 29	100 ± 2	318 ± 23	1325 ± 23	2212 ± 94	733 (4)
50	0.5	100 ± 0	443 ± 60	115 ± 3	391 ± 37	673 ± 12	1156 ± 79	480 (7)
50	0.75	100 ± 0	374 ± 31	153 ± 4	422 ± 40	271 ± 7	535 ± 35	309 (11)
75	0.25	100 ± 0	378 ± 26	101 ± 2	328 ± 24	2780 ± 28	3700 ± 136	1231 (2)
75	0.5	100 ± 0	392 ± 51	111 ± 3	400 ± 37	1362 ± 18	1930 ± 114	716 (5)
75	0.75	100 ± 0	376 ± 28	153 ± 5	433 ± 31	326 ± 6	630 ± 43	320 (9)
100	0.25	100 ± 0	368 ± 22	96 ± 2	298 ± 12	3386 ± 42	4481 ± 80	<b>1455 (1)</b>
100	0.5	100 ± 0	374 ± 41	108 ± 3	361 ± 25	1522 ± 26	2247 ± 122	785 (3)
100	0.75	100 ± 0	329 ± 21	140 ± 4	362 ± 30	333 ± 7	600 ± 30	311 (10)
<i>Mean (rk)</i>		100 (6)	364 (3)	121 (5)	312 (4)	1004 (2)	<b>1667 (1)</b>	
bQAP		(B) The number of calls of $\mathcal{I}$ in a PLS, # $\mathcal{I}$						
25	0.25	143 ± 3	29 ± 13	161 ± 4	33 ± 12	198 ± 6	32 ± 6	99 (9)
25	0.5	64 ± 1	12 ± 6	79 ± 3	16 ± 5	121 ± 8	24 ± 3	53 (11)
25	0.75	23 ± 0	4 ± 2	36 ± 2	7 ± 1	59 ± 6	11 ± 1	23 (12)
50	0.25	491 ± 9	95 ± 45	570 ± 18	118 ± 45	591 ± 10	120 ± 17	331 (5)
50	0.5	205 ± 5	33 ± 17	256 ± 12	48 ± 17	353 ± 17	73 ± 9	162 (7)
50	0.75	59 ± 1	10 ± 5	97 ± 7	18 ± 4	170 ± 18	31 ± 3	64 (10)
75	0.25	1114 ± 18	197 ± 99	1357 ± 20	268 ± 103	1126 ± 15	253 ± 32	720 (2)
75	0.5	457 ± 10	67 ± 38	603 ± 11	101 ± 38	670 ± 7	146 ± 17	341 (4)
75	0.75	98 ± 2	17 ± 8	167 ± 14	31 ± 7	294 ± 30	56 ± 5	110.5 (8)
100	0.25	2021 ± 42	343 ± 166	2331 ± 57	518 ± 180	1739 ± 28	442 ± 53	<b>1232 (1)</b>
100	0.5	758 ± 16	121 ± 68	955 ± 39	189 ± 69	1006 ± 21	239 ± 27	545 (3)
100	0.75	142 ± 3	27 ± 13	244 ± 22	50 ± 10	429 ± 44	89 ± 7	163.5 (6)
<i>Mean (rk)</i>		465 (6)	80 (4)	<b>571 (1)</b>	116 (5)	563 (2)	126 (4)	
bQAP		(C) The number of solutions in the Pareto local optimal set $ A $						
25	0.25	28 ± 6	27 ± 5	27 ± 6	28 ± 5	15 ± 5	19 ± 5	24 (5)
25	0.5	11 ± 4	8 ± 3	10 ± 4	8 ± 3	8 ± 3	7 ± 3	9 (8)
25	0.75	2 ± 1	2 ± 1	2 ± 1	2 ± 1	2 ± 1	2 ± 1	2 (12)
50	0.25	55 ± 10	50 ± 11	55 ± 10	49 ± 11	24 ± 8	29 ± 8	44 (3)
50	0.5	21 ± 6	14 ± 6	21 ± 6	15 ± 6	12 ± 4	11 ± 4	16 (7)
50	0.75	4 ± 2	3 ± 1	4 ± 2	3 ± 2	4 ± 2	3 ± 1	3.5 (10)
75	0.25	90 ± 14	74 ± 16	89 ± 14	76 ± 15	32 ± 11	42 ± 13	67 (2)
75	0.5	35 ± 9	22 ± 9	34 ± 9	24 ± 9	16 ± 6	16 ± 6	24.5 (5)
75	0.75	5 ± 3	3 ± 2	5 ± 3	3 ± 2	5 ± 2	4 ± 2	4 (10)
100	0.25	126 ± 17	102 ± 19	124 ± 19	110 ± 20	37 ± 14	50 ± 18	<b>91.5 (1)</b>
100	0.5	46 ± 11	30 ± 12	46 ± 12	32 ± 12	19 ± 7	19 ± 7	32 (4)
100	0.75	5 ± 3	4 ± 2	6 ± 3	4 ± 3	5 ± 2	4 ± 2	5 (9)
<i>Mean (rk)</i>		<b>36 (1)</b>	28 (4)	35 (2)	29.5 (3)	15 (6)	17 (5)	

There is a natural relationship between Table 3(A), (B) and (C). In Sect. 4.2, we showed that the expected number of solutions generated with  $\mathcal{I}_B$  is much larger than the expected number of solutions generated with  $\mathcal{I}_F$ , which, at its turn, is much larger than the solutions generated with  $\mathcal{I}_N$ . The more often the Pareto improvement strategy  $\mathcal{I}$  is called, the less times SPLS has to restart PLS. Thus,  $\#\mathcal{I}$  and the number of solutions generated with one of the Pareto neighbourhood exploration strategies are inversely correlated with  $\#\text{PLS}$ , while  $\#\mathcal{I}$  is correlated with  $|A|$ . Logically, the larger the size of a Pareto local optimal set, the larger the number of  $\mathcal{I}$  calls that are necessary to explore the neighbourhood. For all algorithms, the largest  $\#\mathcal{I}$  is obtained for bQAP(100, 0.25) which has the largest fronts  $|A|$ .

The least number of times a SPLS restarts PLS is 100 for bMPLS, closely followed by fMPLS. Similar with bMPLS, Paquete and Stützle (2006) restarts PLS 100 times. bGPLS and fGPLS restart up to 3–3.5 times more often. The largest number of times PLS is restarted,  $\#\text{PLS}$ , is obtained with nGPLS, which can be up to 40 times larger than 100. When  $\mathcal{I}_B$  is used,  $\#\text{PLS}$  is independent of  $n$  and  $\rho$  because the expected number of solutions visited with  $\mathcal{I}_B$  is also constant with  $n$ .  $\#\text{PLS}$  of SPLSs using  $\mathcal{I}_N$  varies greatly with  $n$  and  $\rho$  because the expected number of solutions evaluated with  $\mathcal{I}_N$  varies greatly with  $n$  and the number of incomparable and dominating solutions in the neighbourhood. As expected,  $\#\text{PLS}$  when  $\mathcal{I}_F$  is used has values between  $\#\text{PLS}$  using  $\mathcal{I}_B$  and  $\mathcal{I}_N$ .

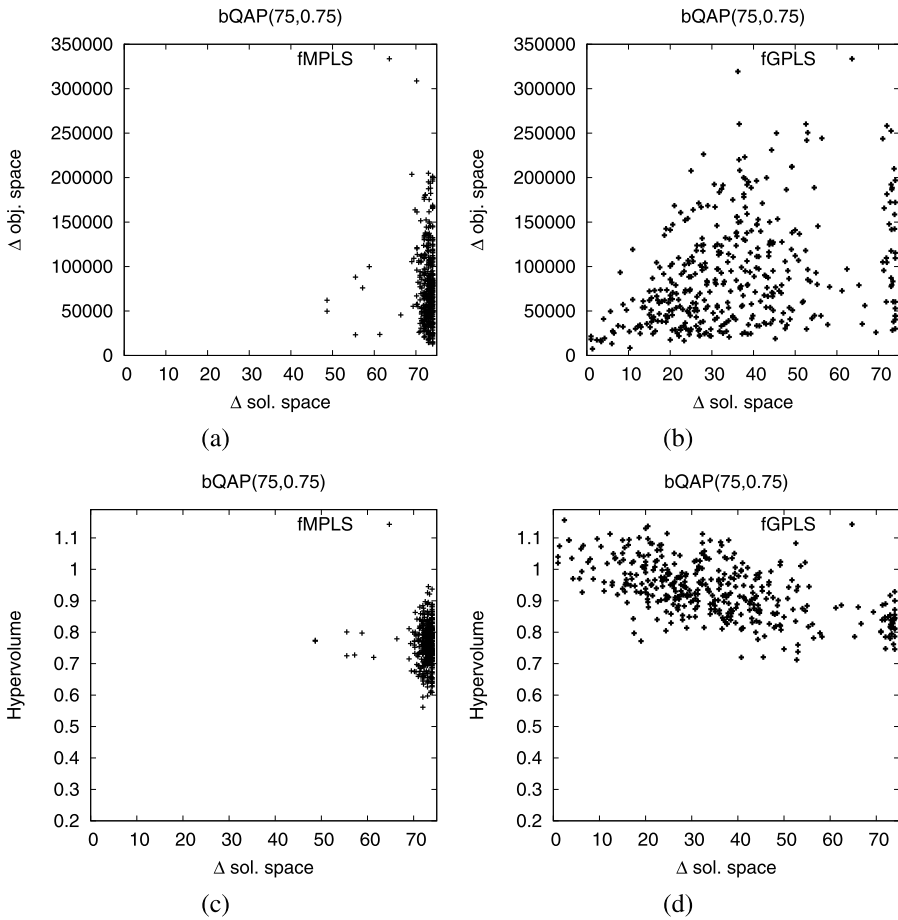
Interesting, for  $\mathcal{I}_N$ , the smaller  $\rho$  is, the larger  $\#\text{PLS}$  is, whereas for  $\mathcal{I}_F$ , the larger  $\rho$  is, the larger  $\#\text{PLS}$  is. We corroborate this observation with the observation that bQAP with lower  $\rho$  have more incomparable solutions, thus larger Pareto local optimal sets. The lower the correlation  $\rho$ , the lower the number of solutions in the neighbourhood that are visited with the  $\mathcal{I}_N$  exploration strategy, and the more often the SPLS using  $\mathcal{I}_N$  has to restart. On the other side, the SPLS using the  $\mathcal{I}_F$  exploration strategy runs longer for low  $\rho$  because of the large number of incomparable solutions in the neighbourhood and the smaller number of dominating solutions that would stop  $\mathcal{I}_F$ .

$\#\mathcal{I}$  and  $|A|$  depend on the type of SPLS, GPLSs have smaller Pareto local optimal sets and smaller  $\#\mathcal{I}$  than MPLSs. The SPLSs using  $\mathcal{I}_N$  have a much smaller  $|A|$  than the SPLSs using  $\mathcal{I}_B$  and  $\mathcal{I}_F$ . This is caused by the numerous incomparable solutions in Pareto local optimal sets of bQAPs with low correlation,  $\rho = 0.25$ . Recall that  $\mathcal{I}_N$  stops after adding a single incomparable solution to  $A$ , whereas  $\mathcal{I}_F$  and  $\mathcal{I}_B$  add a set of solutions to  $A$ .

To conclude, GPLSs spend less time in reaching a Pareto local optimal set than MPLSs do. SPLSs using  $\mathcal{I}_N$  restart most often and they have the worst performance because  $\mathcal{I}_N$  prematurely stops the evaluation of incomparable solutions on large Pareto sets. SPLSs using  $\mathcal{I}_F$  have the best performance and are restarted more often than SPLSs using  $\mathcal{I}_B$ .

### 5.3 Dynamical behaviour of SPLSs on bQAPs

In this section, we empirically analyse the dynamical behaviour of SPLSs using the relationship between Pareto local optimal sets generated at consecutive iterations of PLS. Let us consider any two consecutive iterations of PLS. The Pareto local optimal



**Fig. 5** The correlation of fMPLS on the left and of fGPLS on the right on bQAP(75, 0.75): (top) between their distance in objective space of consecutively restarted PLSs and their distance in solution space, (bottom) between the hypervolume of restarted PLSs and their average distance in solution space

set of the first PLS is denoted as *the parent Pareto local optimal set*, or shortly *the parent Pareto set*. The output of the second PLS is denoted as *the offspring Pareto local optimal set*, or shortly *the offspring Pareto set*. The distance in the objective space,  $\Delta_{obj}$ , between any parent and its offspring Pareto set is calculated as the average Euclidean distance between any two solutions, one from the parent and one from its child. The distance in the solution space,  $\Delta_{sol}$ , between a parent and its offspring Pareto set is calculated as the average of the minimum number of swaps between any two solutions of the two Pareto local optimal sets. For this experiment, Pareto local optimal sets of a single random run of SPLSs are considered.

Figure 5(a) and (b) show, on bQAP(75, 0.75), the relationship between  $\Delta_{obj}$  and  $\Delta_{sol}$ . Figure 5(c) and (d) present, on bQAP(75, 0.75), the relationship between the distance in solution space,  $\Delta_{sol}$  and the hypervolume of the offspring Pareto set. Note the resemblance between the figures for the same SPLS. For fMPLS, solutions

**Table 4** The distance between parent and offspring in (A) solution space,  $\Delta sol$ , and (B) objective space,  $\Delta obj$ 

$n$	$\rho$	bMPLS	bGPLS	fMPLS	fGPLS	nMPLS	nGPLS
bQAP		(A) The distance between parent and offspring Pareto local optimal sets in solution space, $\Delta sol$					
25	0.25	20 ± 0	17 ± 1	20 ± 0	17 ± 1	20 ± 0	19 ± 1
25	0.5	20 ± 0	12 ± 2	20 ± 0	14 ± 2	20 ± 0	16 ± 1
25	0.75	18 ± 1	5 ± 3	18 ± 1	8 ± 3	19 ± 1	10 ± 2
50	0.25	43 ± 0	31 ± 3	43 ± 0	32 ± 4	43 ± 0	40 ± 1
50	0.5	43 ± 0	22 ± 5	43 ± 0	26 ± 4	43 ± 0	32 ± 3
50	0.75	41 ± 1	13 ± 4	42 ± 1	19 ± 4	42 ± 1	25 ± 3
75	0.25	66 ± 0	42 ± 5	66 ± 0	47 ± 3	66 ± 0	61 ± 3
75	0.5	66 ± 0	32 ± 7	66 ± 0	41 ± 6	66 ± 0	52 ± 6
75	0.75	64 ± 1	21 ± 5	64 ± 1	31 ± 4	65 ± 1	40 ± 5
100	0.25	89 ± 0	67 ± 0	88 ± 0	62 ± 4	88 ± 0	80 ± 0
100	0.5	88 ± 0	44 ± 4	88 ± 0	55 ± 8	88 ± 0	80 ± 1
100	0.75	87 ± 1	34 ± 7	87 ± 0	48 ± 5	88 ± 1	58 ± 6
bQAP		(B) The distance between parent and offspring Pareto local optimal sets in objective space $\times 10^3$ , $\Delta obj$					
25	0.25	51 ± 5	65 ± 9	50 ± 5	67 ± 7	48 ± 5	74 ± 11
25	0.5	32 ± 3	30 ± 5	31 ± 3	32 ± 4	35 ± 6	37 ± 5
25	0.75	17 ± 4	9 ± 5	18 ± 5	13 ± 5	22 ± 7	19 ± 6
50	0.25	136 ± 17	161 ± 23	134 ± 16	163 ± 19	139 ± 25	181 ± 22
50	0.5	78 ± 6	73 ± 10	79 ± 7	78 ± 9	92 ± 25	89 ± 10
50	0.75	43 ± 8	29 ± 9	44 ± 9	41 ± 11	59 ± 30	53 ± 12
75	0.25	233 ± 24	285 ± 21	233 ± 9	276 ± 15	266 ± 83	303 ± 45
75	0.5	131 ± 11	121 ± 16	135 ± 16	132 ± 16	175 ± 58	156 ± 24
75	0.75	73 ± 13	59 ± 19	70 ± 12	80 ± 17	105 ± 40	98 ± 23
100	0.25	360 ± 15	386 ± 0	418 ± 0	398 ± 43	467 ± 40	518 ± 0
100	0.5	182 ± 20	181 ± 27	184 ± 13	186 ± 15	278 ± 97	252 ± 29
100	0.75	103 ± 22	102 ± 19	106 ± 25	135 ± 25	171 ± 74	160 ± 35

in parent and offspring Pareto sets are at a large distance in solution space and they are uncorrelated with each other. For fGPLS, solutions in parent and offspring Pareto set are correlated. A small distance in solution space corresponds to a small distance in the objective space and a high hypervolume indicator for the offspring Pareto set. A large distance in solution space corresponds to a large distance in objective space and a lower hypervolume indicator for the offspring.

In Table 4, we calculate: (i)  $\Delta obj$  in Table 4(A), and (ii)  $\Delta sol$  in Table 4(B). As expected, Table 4 generalizes the results from Fig. 5. For MPLSs, the distance in solution space,  $\Delta sol$ , is proportional with  $n$  and is independent of  $\rho$  and  $\mathcal{I}$ . For GPLSs,  $\Delta sol$  is proportional with  $n$  and inverse proportional with  $\rho$ .  $\Delta sol$  has the

smallest values for GPLSs using  $\mathcal{I}_B$ , and the largest values for GPLSs using  $\mathcal{I}_N$ .  $\Delta_{sol}$  for GPLSs is smaller than for MPLSs.

$\Delta_{obj}$  is proportional with  $n$  and inverse proportional with  $\rho$ . For GPLS, a large distance in objective space  $\Delta_{obj}$  implies also a large distance in solution space  $\Delta_{sol}$ .  $\Delta_{obj}$  has the smallest values for SPLSs using  $\mathcal{I}_B$ , and the largest values for SPLSs using  $\mathcal{I}_N$ .

To conclude, genetic PLSs perform better than multi-restart PLSs by exploiting the structure of the bQAP search space, a structure caused by the positive correlation between the objectives. As shown above, the distance in solution space is correlated with the distance in objective space between consecutive Pareto local optimal sets and with the hypervolume indicator.

## 6 Random walks for landscape analysis of bQAPs using SPLSs

The previous section showed that the best performance was obtained with the SPLS using genetic perturbation operators and the first Pareto improvement neighbourhood exploration strategy. We presented some experimental results where GPLSs exploit the correlation in bQAPs. In this section, we deepen our analysis using random walks to investigate the effects of various mechanisms of stochastic PLSs in isolation. A *random walk* (RW) is a sequence of steps generated with a random process on the path between two “good” solutions in the search space using a genetic operator.

Section 6.1 shows the usefulness of the genetic operators that use cycles. The efficiency of *deactivation* for different correlation values of bQAPs is studied in Sect. 6.2. A novel landscape analysis with RW for multi-objective spaces is given in Sect. 6.3. The properties of different Pareto neighbourhood exploration strategies are investigated with RW in Sect. 6.4.

*Methodology* Let  $s$  and  $s'$  be two solutions selected uniform randomly without replacement from the best known Pareto local optimal set  $A^*$  of a bQAP instance. Let  $\Delta(s, s')$  be the distance between the two solutions and let  $s_1$  be a solution generated on the path between two solutions  $s$  and  $s'$ . In this experiment,  $\Delta(s, s') = 70$ . For each value of  $q \in \{1, 2, \dots, \Delta(s, s')\}$ , 100 solutions are generated by swapping  $q$  pairs of facilities with each genetic operator.

Using random walks, we study the relationship between the quality of the solutions generated on the path at a certain distance from  $s$  and the quality of the Pareto local optimal set of PLSs started from these solutions. We consider two initial Pareto sets  $A$  for PLS( $A, \mathcal{I}$ ): (a)  $A = \{s_1\}$ , which does not consider deactivation, and (b)  $A = \text{Deactiv}(s_1, A^*)$  which considers deactivation. By definition,  $\text{Deactiv}(s_1, A^*)$  contains  $s_1$  and all the solutions in  $A^*$  that are incomparable with  $s_1$ . For the ease of the discussion, we have set the visited flag of the solutions from  $A^*$  included in  $\text{Deactiv}(s_1, A^*)$  to true. When no deactivation is used, the exploratory archive is empty and solutions are only compared to the newly generated solution  $s$ . Later, the newly found solutions are merged with the complete archive, so that only non-dominated solutions remain. Another alternative would be to restart PLS from a set  $\text{merge}(\{s_1\}, A^*)$ . Preliminary experiments showed a very poor performance of such an algorithm because

the probability that  $\text{merge}(\{s_1\}, A^*) \neq A^*$  is very small. Most of the time, such PLS is stuck in the same, early found, Pareto local optimum set.

The quality of a Pareto local optimal set is measured with the hypervolume unary indicator computed as indicated in Sect. 5. The objectives of all generated Pareto local optimal sets, including the objectives in  $A^*$ , are normalized, each objective having values between 1 and 2. The reference solution for the hypervolume unary indicator is, as before, (2.1, 2.1). A large hypervolume value means a better Pareto local optimal set. A PLS is restarted for each of the 100 points generated at certain distances on the path.

### 6.1 Cycle vs non-cycle based genetic operators for bQAPs

It is important to recall that the perturbation operators we used in Sect. 4.1, mutation and path-guided mutation, preserve the cycles in the parents. To motivate our choice, we show here that these operators are more efficient than perturbation operators that do not respect cycles. The efficiency is measured with the hypervolume unary indicator and the Euclidean distance in objective space. In Table 2, the largest difference between GPLSs and MPLSs are for bQAPs with  $\rho = 0.75$ . We focus here on this test problem.

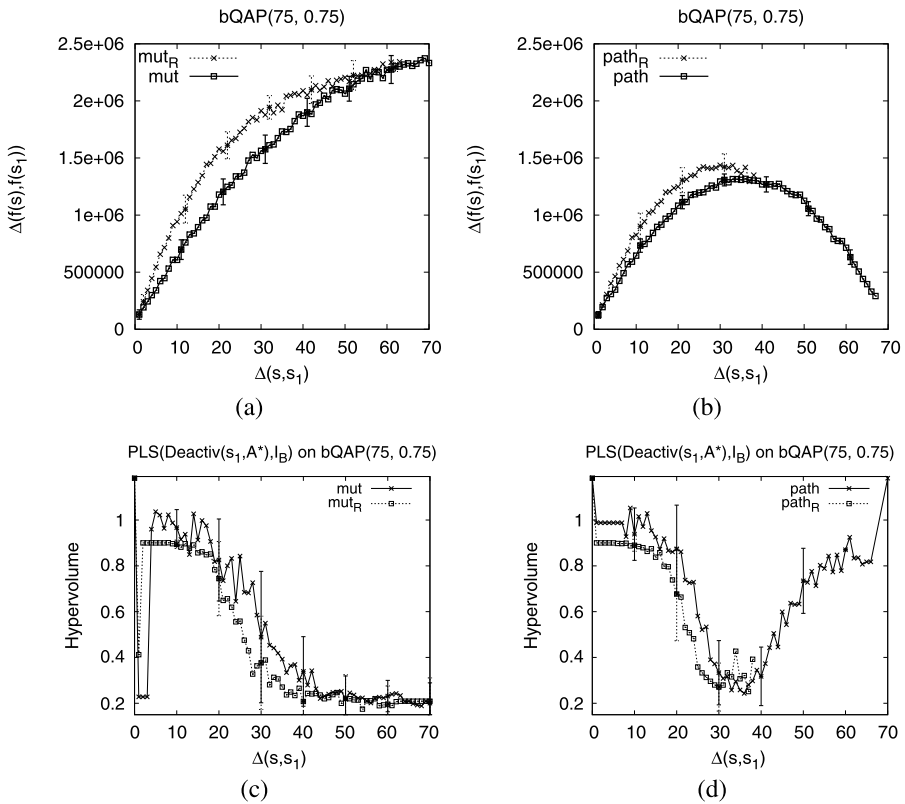
We compare four operators. The two operators that respect cycles are: (i) the  $q$ -exchange mutation,  $mut$ , and (ii) the  $q$ -exchange path-mutation,  $path$ . We also consider two operators, mutation  $mut_R$  and path-guided mutation  $path_R$ , that do not respect cycles.  $mut_R$  swaps  $q$  uniform randomly chosen pairs of facilities, with replacement.  $path_R$  swaps  $q$  pairs of facilities such that, for each swap: (i) the first location,  $i$ , is uniform randomly chosen from a parent  $s$ , and (ii) the second location,  $j$ , is chosen to have the same value in  $s'$  like the value in the  $i$ -th location in  $s$ .

In Fig. 6, the  $x$ -axis represents the distance in solution space between the initial solution  $s$  and the solution  $s_1$ ,  $\Delta(s, s_1)$  generated with one of the four operators. Note that for the operators that do respect cycles,  $mut$  and  $path$ , we have  $\Delta(s, s_1) = q$ . For operators that do not respect cycles,  $mut_R$  and  $path_R$ , we have  $\Delta(s, s_1) \leq q$ . To see this, one should note that by randomly sampling positions for pairing, some positions might be swapped two or more times. In this case the distance between the generated and the initial solution does not increase. In Fig. 6(a) and (b), the  $y$ -axis denotes the Euclidean distance between the objectives of  $s$  and  $s_1$ ,  $\Delta(f(s), f(s_1))$ . In Fig. 6(c) and (d), the  $y$ -axis denotes the hypervolume unary indicator for a Pareto local optimal set generated by  $\text{PLS}(\text{Deactiv}(s_1, A^*), \mathcal{I}_B)$  from Algorithm 1. The hypervolume indicator value for  $A^*$  is shown at the borders in Fig. 6(c) and (d).

In Fig. 6(a) and (c),  $mut$  generates solutions that are closer in objective space to the initial solution,  $s$ , than the solutions generated with  $mut_R$ . The difference between the two operators, in both objective space and hypervolume indicators, is diminished for large  $\Delta(s, s_1)$ , i.e. 50 to 70 swaps. Note that for a large amount of swaps,  $s_1$  is actually randomly generated. The largest difference between the two operators is for 20 to 30 swaps. This indicates a better exploration of the structure of the problem for  $mut$  than with  $mut_R$ . Wilcoxon non-parametric two-side tests showed that these differences in hypervolume are significant. We conclude that  $mut_R$  is worse than  $mut$ .

In Fig. 6(b) and (d),  $path$  generates solutions that are closer in objective space to  $s$  and  $s_1$  than solutions generated with  $path_R$ . Similarly with mutation, the largest



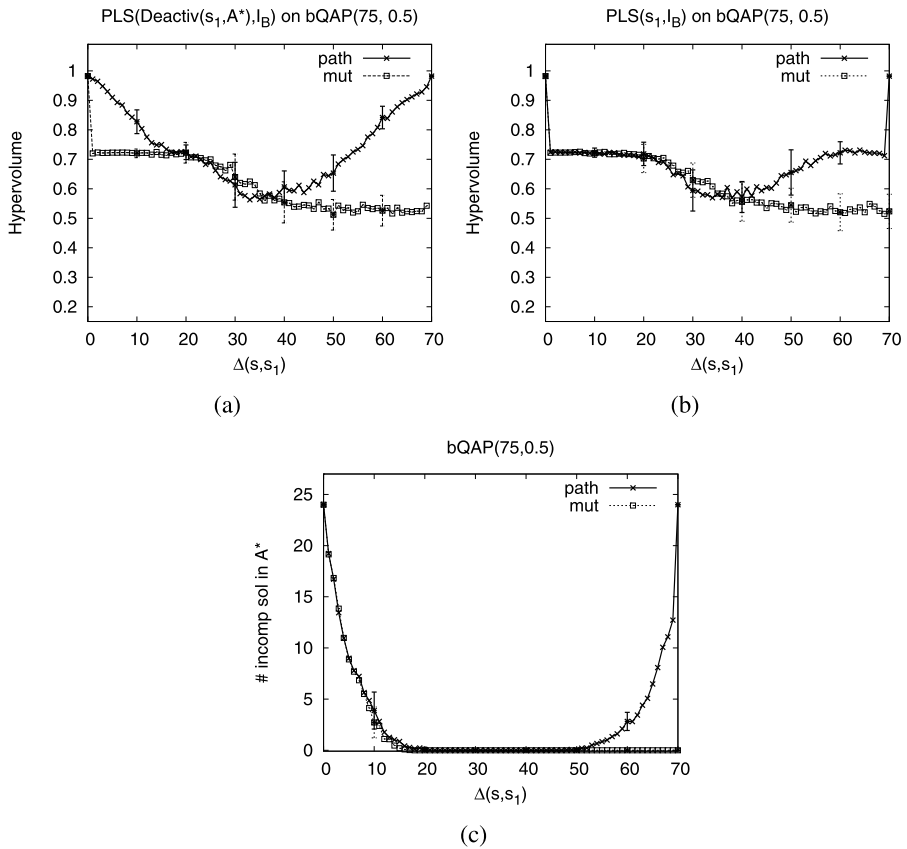


**Fig. 6** Random walk (RW) between two solutions  $s$  and  $s'$  from  $A^*(75, 0.75)$  for four genetic operators (a) *mut* and *mut<sub>R</sub>* and (b) *path* and *path<sub>R</sub>*. The top part shows the Euclidean distance in objective space between the initial solution  $s$  and its offspring  $s_1$  on the path towards  $s'$ ,  $\Delta(f(s), f(s'))$ . The bottom part shows the hypervolume of a Pareto local optimal set of PLSs that are started from the solutions generated from  $s_1$  on the path between two solutions,  $s$  and  $s'$ ,  $PLS(Deactiv(s_1, A^*), \mathcal{I}_B)$  using (c) *mut* and *mut<sub>R</sub>*, and (d) *path* and *path<sub>R</sub>*

difference between *path* and *path<sub>R</sub>* is between 20 and 30 swaps which coincides with the largest and most significant difference between the corresponding hypervolume indicators. Note that, for *path<sub>R</sub>*, there are no solutions  $s_1$  generated at large distances from  $s$ ,  $\Delta(s, s_1) > 40$ .

Let us compare mutation *mut* and path-guided mutation *path*. The solutions that are far-away from the first parent in *path* are, by definition, close to the second parent and thus bounded. For  $\Delta(s, s_1) > 35$ , the solutions generated with *path* have lower Euclidean distances in objective space than the solutions generated with *mut*. The corresponding hypervolume unary indicator for *path* is significantly larger than for *mut*. Thus, the path-guided mutation is better than mutation in exploring the structure of the landscape. Using the same arguments, *path<sub>R</sub>* is better than *mut<sub>R</sub>*.

To conclude, the operator that best explores the structure of the search space is *path* which preserves both the cycles and walks on the path between two good solutions. It is interesting to note the relationship between the hypervolume indicator of a Pareto



**Fig. 7** The Pareto local optimal set of PLSs started from  $s_1$  on the path between two solutions in  $A^*(75, 0.5)$ : (a) using the deactivation technique  $PLS(Deactiv(s_1, A^*), \mathcal{I}_B)$ , (b) without deactivation  $PLS(s_1, \mathcal{I}_B)$ . (c) The number of solutions in  $A^*(75, 0.5)$  that are incomparable with the solutions generated with *mut* and *path* on the path between  $s$  and  $s'$

local optimal set of PLSs restarted from a solution  $s_1$  of the path and the distance from the initial solution  $s$  to  $s_1$ . This is a particularity of bQAP’s landscape captured also in Fig. 5. It basically means that solutions that are correlated in solution space are also correlated in objective space. That is why genetic PLSs are so much better than the multi-restart PLS for highly correlated bQAPs.

### 6.2 The use of deactivation in bQAPs

In this section, we measure the effect of the *deactivation* technique on PLS’s performance. In Fig. 7(a) and (b), we compare the hypervolume indicators of PLSs restarted from  $s_1$  on the path between two solutions  $s$  and  $s'$  from  $A^*(75, 0.5)$ . In Fig. 7(c), we count the solutions from  $A^*(75, 0.5)$  that are incomparable with the generated solution  $s_1$ .

The hypervolume indicator for PLSs restarted with solutions generated with *path* and using the deactivation technique are the largest for  $\Delta(s, s_1) \in \{1, \dots, 15\} \cup$

{55, . . . , 69}. For the same distances, there is a positive number of solutions, maximum 18 from a total of 24, that are incomparable with  $s_1$ . For *path*, the increase in the quality of Pareto local optimal sets generated with PLS restarted in  $s_1$  is reflected in the increased number of incomparable solutions in  $A^*$ . For *mut*, when  $s_1$  is such that  $\Delta(s, s_1) \in \{1, \dots, 15\}$ , the number of incomparable solutions is as large as for solutions generated with *path*. For a large number of swaps, the number of incomparable solutions in  $A^*$  is 0 indicating a rapid decrease in quality of the generated solutions. Note that for solutions generated with *mut* the deactivation technique does not increase the quality of the output of the restarted PLSs.

Note the difference in the hypervolume indicator of bMPLS from Table 2(A) and the hypervolume indicators for PLS restarted at a large distance from  $s$  from a solution  $s_1$  generated with *mut* from Fig. 6(c) and (d), and Fig. 7(a) and (b). In this section, to compute the hypervolume indicator, we normalize all the Pareto local optimal sets of PLSs restarted on the path and  $A^*$  of a bQAP. In the previous section, we normalized all the Pareto local optimal sets of the tested SPLSs. Some of the solutions in these Pareto sets are dominating solutions in  $A^*$ , and some of these solutions are much worse than the Pareto local optimal sets from the experiments in this section.

For bQAP(75, 0.75), for both *mut* and *path*, there are few solutions, at most 2 from a total number of solutions of 11, that are incomparable with the current solution  $s_1$  in  $A^*(75, 0.75)$ , where  $\Delta(s, s_1) \in \{1, 2\}$ . For the rest of solutions  $s_1$  on the path,  $\Delta(s, s_1) > 2$ , there are no incomparable solutions in  $A^*$ .

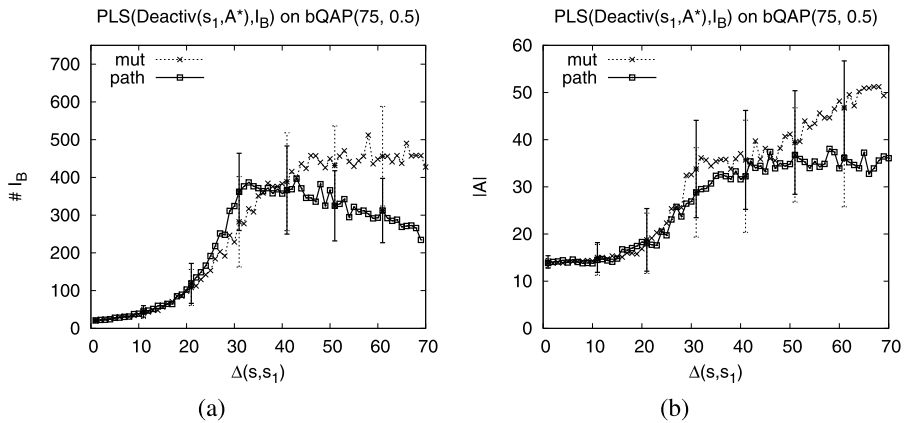
In conclusion, for  $\rho = 0.75$ , the size of Pareto local optimal sets are small and the deactivation mechanism does not work. For lower correlations,  $\rho = 0.5$ , the size of Pareto local optimal sets are larger and the deactivation selects several incomparable solutions, which improves results a lot with the path-guided mutation operator.

### 6.3 The properties of PLSs restarted from RW solutions

In this subsection, we investigate the properties of PLSs that are started from the solutions on the path between  $s$  and  $s'$ ,  $s, s' \in A^*$ , using one of the genetic operators *mut* and *path*. Consider PLS(Deactiv( $s_1, A^*$ ),  $\mathcal{I}_B$ ) and bQAP(75, 0.5) as before.

Let us measure how many times in a PLS run the Pareto improvement strategy  $\mathcal{I}_B$  is called. The number of swaps in the PLS run is the number of calls of  $\mathcal{I}_B$  multiplied with the number of solutions in a neighbourhood, i.e.  $C_2^n$  with  $n$  the number of facilities. Figure 8(a) shows the relationship between  $\Delta(s, s_1)$  and  $\#\mathcal{I}_B$ , the number of calls of  $\mathcal{I}_B$  in the PLS started from  $s_1$ . For *mut*,  $\Delta(s, s_1)$  is proportional with  $\#\mathcal{I}_B$ . For *path*, the largest  $\#\mathcal{I}_B$  is obtained for  $\Delta(s, s_1) = \Delta(s, s')/2$ . In Table 3(B), for bQAP(0.75, 0.5),  $\#\mathcal{I}_B$  for bMPLS is 457 that is approximately the same as  $\#\mathcal{I}_B$  for PLSs restarted using *mut* at a large distance from  $s$ . The number of restarts  $\mathcal{I}_B$  for bGPLS is 67 which corresponds to small  $\Delta(s, s_1) \in \{10, \dots, 20\}$ . Recall that the exchange rates for mutation and path-guided mutation for GPLSs on bQAP(75, 0.5) are uniform randomly selected from  $\{3, \dots, 38\}$ . The number of restarts in bGPLS cannot be directly related with  $\#\mathcal{I}_B$  from this experiment.

Let us compare  $\mathcal{I}_B$  from Fig. 8(a) with the hypervolume indicators from Fig. 7(a). A small number of calls of  $\mathcal{I}_B$  corresponds with high hypervolume unary indicators. A large  $\#\mathcal{I}_B$  corresponds with low hypervolume unary indicator. This means that



**Fig. 8** Consider the Pareto local optimal set of PLSs started from  $s_1$  on the path between two solutions in  $A^*(75, 0.5)$ ,  $PLS(Deactiv(s_1, A^*), \mathcal{I}_B)$ . **(a)** The number of times  $\mathcal{I}_B$  is called,  $\# \mathcal{I}_B$ . **(b)** The number of solutions in the Pareto local optimal sets,  $|A|$

randomly restarting PLS is outperformed by SPLSs that are restarted from solutions close in solution and objective space to their parent solutions.

In Fig. 8(b), the size of a Pareto local optimal set,  $|A|$  between 10 and 50 solutions, is increasing with the increase of  $\Delta(s, s_1)$ . Again, for a large  $\Delta(s, s_1)$ ,  $|A|$  is larger for solutions generated with *mut* than for solution generated with *path*. This relationship is found back in Table 3(C) for bQAP(75, 0.5) where  $|A|$  for bMPLS is 35 and for bGPLS is 22. This also means that larger archives do not always correspond to better Pareto local optimal sets.

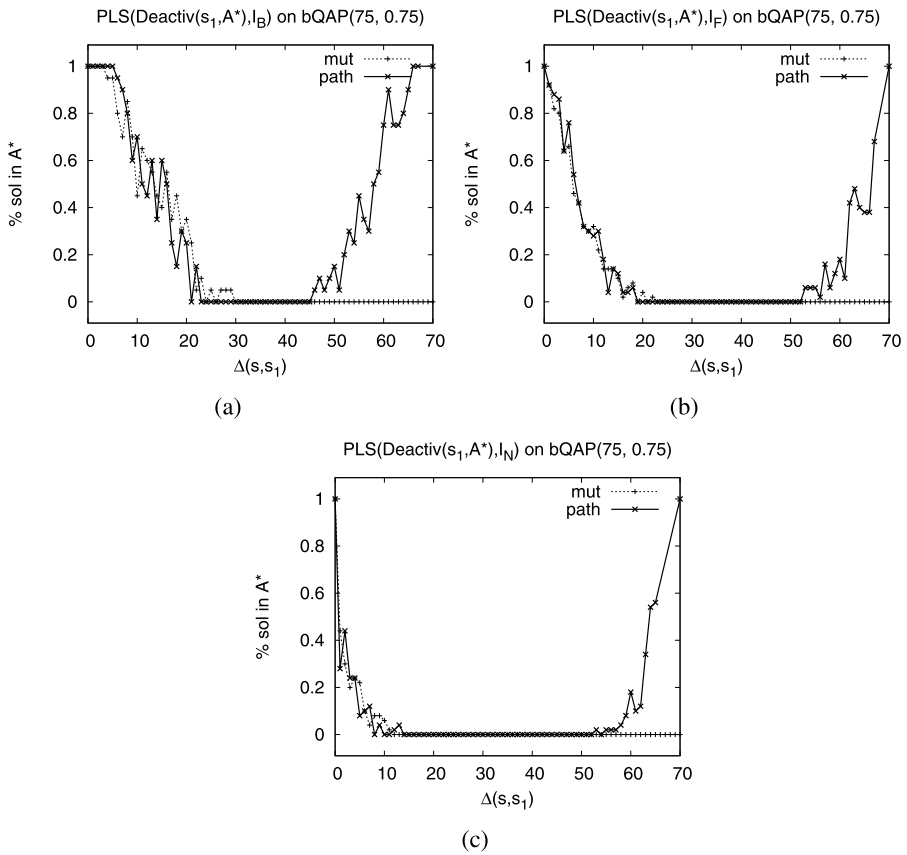
To conclude, the results obtained with RW are closely related to the results in Sect. 5, meaning that RW can be used in the context of SPLS to study the performance of these algorithms.

### 6.4 RW landscape analysis for different Pareto neighbourhood exploration strategies

In this section we compare the properties of the three Pareto neighbourhood exploration strategies,  $\mathcal{I}_B$ ,  $\mathcal{I}_F$  and  $\mathcal{I}_N$ , on a bQAP instance bQAP(75, 0.75). We show that the basins of attraction are explored differently by the three Pareto neighbourhood exploration strategies.

In Fig. 9(a), if  $s_1$  is generated with *mut* such that  $\Delta(s, s_1) \geq 30$ , PLS using  $\mathcal{I}_B$  escapes the basin of attraction of  $s$ . When *path* is used, the probability to find  $s'$ , the second parent of  $s_1$ , in PLS's output is non-zero if the distance between  $s_1$  and  $s'$ ,  $\Delta(s_1, s')$ , is lower than 25. In Fig. 9(b) and (c), PLSs that use  $\mathcal{I}_F$  and  $\mathcal{I}_N$  explore less well the basins of attraction of  $s$  and  $s'$  than  $\mathcal{I}_B$  does. The probability to return in  $s$  with  $\mathcal{I}_F$  is non-zero for  $\Delta(s, s_1) < 20$  and it is smaller than with  $\mathcal{I}_B$ . This is because the search is stopped before all the solutions in the neighbourhood are explored. As expected, PLSs using  $\mathcal{I}_F$  explore better the basins of attraction than PLSs using  $\mathcal{I}_N$ .

To conclude, note that the “optimal” number of exchanges in the two genetic operators is always a trade-off between the probability of leaving the basin of attraction, and the probability of generating a new local optimum that is closest to—and thus



**Fig. 9** The probability that a Pareto local optimal set of PLSs started on the path between two solutions in  $A^*(75, 0.75)$  contains the parent solutions. PLS uses one of the following Pareto neighbourhood exploration strategies: (a)  $I_B$ , (b)  $I_F$ , and (c)  $I_N$

most correlated with—the current Pareto local optimal set. For a small number of exchanges, the solutions generated are “fit” in the sense that the objective values approach the objective value of the initial solution and the hypervolume of the Pareto local optimal sets of PLS restarted from these solutions is high. The smaller the number of exchanges, the higher the probability that solutions belong to the basin of attraction of the initial solution.

### 7 Conclusions and discussion

We introduced *stochastic PLS* (SPLS), a class of algorithms that combine Pareto local search with stochastic perturbation operators to efficiently explore multi-objective search problems. The paper focuses on three issues: (i) Pareto neighbourhood exploration strategies to explore the Pareto neighbourhood of a solution, (ii) stochastic perturbation operators, mutation and path-guided mutation, to restart the Pareto local

search, and (iii) the deactivation method to run PLS from perturbed solutions in the context of previous runs of Pareto local search.

We proposed a new concept for Pareto neighbourhood exploration strategies that compares the solutions in a neighbourhood with: (a) the current solution from a Pareto set, and (b) all the other solutions in that Pareto set which were previously generated with Pareto local search. We show that our Pareto improvement strategies, especially neutral and first Pareto improvement, improve the performance of similar strategies from the literature (Liefvooghe et al. 2011). Interestingly, we showed that the use of the neutral Pareto improvement strategy with PLS is equivalent to the use of the first improvement strategy with hypervolume based local search. The hypervolume unary indicator is widely used to transform a multi-objective problem into a single-objective problem (Beume et al. 2009). However, our experiments indicate that SPLSs that use the neutral Pareto improvement—or equivalently, hypervolume based local search using the first improvement exploration strategy—perform the worst especially for bQAPs with a large number of incomparable solutions because of the lack of exploration around the evaluated solutions. SPLSs that use the first Pareto improvement strategy have the highest performance.

We presented genetic operators, mutation and path-guided mutation, specifically designed for multi-objective QAPs. We used random walks to show that our genetic operators that use cycles efficiently exploit this search space. We also added explanations on why some algorithms work better than the others. Overall, our experimental results showed that genetic PLSs outperform multi-restart PLSs for bi-objective QAPs that have a structured search space that is exploitable with genetic operators—in our case positive correlation between objectives.

In the experimental section, we explained the difference in behaviour of six different (S)PLS algorithms using some characteristics of the landscape. Random walks were successfully used to analyse the fitness landscape and PLS's behaviour. We showed that the deactivation method improves the performance of PLSs for bQAPs with large Pareto local optimal sets, thus with a large amount of incomparable solutions. To make the results in this paper comparable with the results from Paquete's study, we restarted bMPLS the same number of times as Paquete did. Some of our SPLS algorithms performed better than Paquete's algorithm, better even than the tabu search algorithm which they run much longer in order to generate a reference set. The best performance with SPLS is obtained using the first Pareto improvement strategy, which stops the neighbourhood exploration when a dominating solution is found, and using stochastic perturbation operators which exploit the particularities of the search space. All the novel techniques proposed in this paper contributed to the efficiency of the SPLS algorithm.

To conclude, we can state that stochastic Pareto local search (SPLS) is a powerful framework for multi-objective, combinatorial optimization problems. Although we only showed experimental results on a single problem type (mQAPs), one can safely predict that SPLS will also be successful for other types of problems, just like stochastic local search (SLS) algorithms are successfully applied in a very large range of single-objective combinatorial optimization problems.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

## References

- Aguirre, H.E., Tanaka, K.: Random bit climbers on multiobjective MNK-landscapes: effects of memory and population climbing. *IEICE Trans. Electron.* **88-A**(1), 334–345 (2005)
- Alsheddy, A., Tsang, E.: Guided Pareto local search and its application to the 0/1 multi-objective Knapsack problem. In: *Metaheur. Inter. Conf. (MIC)* (2009)
- Angel, E., Bampis, E., Gourvès, L.: A dynasearch neighbourhood for the bicriteria traveling salesman problem. In: *LNEMS*, vol. 535, 153–176 (2004)
- Bader, J.: Hypervolume-Based search for multiobjective optimization: theory and methods. Ph.D. thesis, ETH, Zurich, Switzerland (2010)
- Bader, J., Zitzler, E.: Hype: an algorithm for fast hypervolume-based many-objective optimization. *Evol. Comput.* **19**(1), 45–76 (2011)
- Basseur, M.: Design of cooperative algorithms for multi-objective optimization: application to the flow-shop scheduling problem. *4OR* **4**, 255–258 (2006)
- Beume, N., Fonseca, C.M., López-Ibáñez, M., Paquete, L., Vahrenhold, J.: On the complexity of computing the hypervolume indicator. *IEEE Trans. Evol. Comput.* **13**, 1075–1082 (2009)
- Beume, N., Naujoks, B., Emmerich, M.: Sms-emoa: multiobjective selection based on dominated hypervolume. *Eur. J. Oper. Res.* **181**, 1653–1669 (2007)
- Bleuler, S., Laumanns, M., Thiele, L., Zitzler, E.: PISA—a platform and programming language independent interface for search algorithms. In: *Proc. of Evol Multi-Criterion Optim. (EMO)*. LNCS, pp. 494–508. Springer, Berlin (2003)
- Bringmann, K., Friedrich, T.: An efficient algorithm for computing hypervolume contributions. *Evol. Comp.* (2010). doi:[10.1162/EVCO\\_a\\_00012](https://doi.org/10.1162/EVCO_a_00012)
- Deb, K., Mohan, M., Mishra, S.: Evaluating the  $\epsilon$ -domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions. *Evol. Comput.* **13**, 501–525 (2005)
- Drugan, M.M., Thierens, D.: Generalized adaptive pursuit algorithm for genetic Pareto local search algorithms. In: *GECCO*, pp. 1963–1970 (2011)
- Drugan, M.M., Thierens, D.: Path-guided mutation for stochastic Pareto local search algorithms. In: *Parallel Problem Solving from Nature (PPSN)*, pp. 485–495. Springer, Berlin (2010)
- Dubois-Lacoste, J., Lopez-Ibanez, M., Stützle, T.: A hybrid tp+pls algorithm for bi-objective flow-shop scheduling problems. *Comput. Oper. Res.* **38**(8), 1219–1236 (2011a)
- Dubois-Lacoste, J., Lopez-Ibanez, M., Stützle, T.: Pareto local search algorithms for anytime bi-objective optimization. Technical report, IRIDIA—Université Libre de Bruxelles (2011b). doi:[10.1007/978-3-642-29124-1\\_18](https://doi.org/10.1007/978-3-642-29124-1_18)
- Emmerich, M., Beume, N., Naujoks, B.: An EMO algorithm using the hypervolume measure as selection criterion. In: *EMO*, pp. 62–76 (2005)
- Fonseca, C.M., Fleming, P.J.: On the performance assessment and comparison of stochastic multiobjective optimizers. In: *Parallel Problem Solving from Nature (PPSN)*, pp. 584–593 (1996)
- Goldberg, D.E., Lingle, R.: In: *Alleles, Loci and the Traveling Salesman Problem* pp. 154–159 (1985)
- Hamacher, H.W., Ruhe, G.: On spanning tree problems with multiple objectives. *Ann. Oper. Res.* **52**, 209–230 (1994)
- Hansen, P., Mladenović, N.: First vs. best improvement: an empirical study. *Discrete Appl. Math.* **154**, 802–817 (2006)
- Jaszkiewicz, A., Zielniewicz, P.: Pareto memetic algorithm with path relinking for bi-objective traveling salesperson problem. *Eur. J. Oper. Res.* **193**, 885–890 (2009)
- Knowles, J.D., Corne, D.W.: Instance generators and test suites for the multiobjective quadratic assignment problem. In: *Proc. of Evol Multi-Criterion Optim (EMO)*, pp. 295–310 (2003)
- Knowles, J.D., Corne, D.W.: Bounded Pareto archiving: theory and practice. In: *Metaheuristics for Multi-objective Optimisation. Lecture Notes in Economics and Mathematical Systems*, vol. 535, pp. 39–64 (2004)
- Liefooghe, A., Humeau, J., Mesmoudi, S., Jourdan, L., Talbi, E.-G.: On dominance-based multiobjective local search: design, implementation and experimental analysis on scheduling and traveling salesman problems. *J. Heuristics* 1–36 (2011). doi:[10.1007/s10732-011-9181-3](https://doi.org/10.1007/s10732-011-9181-3)
- Lopez-Ibanez, M., Knowles, J.D., Laumanns, M.: On sequential online archiving of objective vectors. In: *Evolutionary Multi-Criterion Optimization (EMO 2011)*. LNCS, vol. 6576, pp. 46–60. Springer, Heidelberg (2011)
- Lopez-Ibanez, M., Paquete, L., Stützle, T.: On the design of aco for the biobjective quadratic assignment problem. In: *Ant Colony Optimization and Swarm Intelligence, 4th International Workshop, ANTS 2004*. LNCS, vol. 3172, pp. 214–225. Springer, Heidelberg (2004)

- Lopez-Ibanez, M., Paquete, L., Stützle, T.: Hybrid population-based algorithms for the bi-objective quadratic assignment problem. *J. Math. Model. Algorithms* **5**, 111–137 (2006)
- Lourenco, H.R., Martin, O.C., Stützle, T.: Iterated local search. In: *Handbook of Metaheur.* LNCS, pp. 320–353. Springer, Berlin (2003)
- Lust, T., Teghem, J.: Two-phase Pareto local search for the biobjective traveling salesman problem. *J. Heuristics* **16**, 475–510 (2010)
- Merz, P., Freisleben, B.: Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. *IEEE Trans. Evol. Comput.* **4**, 337–352 (2000)
- Paquete, L., Chiarandini, M., Stützle, T.: Pareto local optimum sets in the biobjective traveling salesman problem: an experimental study. In: *Metaheur. for Multiobj. Optim.*, vol. 535, pp. 177–200. Springer, Berlin (2004)
- Paquete, L., Schiavinotto, T., Stützle, T.: On local optima in multiobjective combinatorial optimization problems. *Ann. Oper. Res.* **156**, 83–97 (2007)
- Paquete, L., Stützle, T.: A study of stochastic local search algorithms for the biobjective QAP with correlated flow matrices. *Eur. J. Oper. Res.* **169**, 943–959 (2006)
- Poon, P.W., Carter, J.N.: Genetic algorithm crossover operators for ordering applications. *Comput. Oper. Res.* **22**, 135–147 (1995)
- Schiavinotto, T., Stützle, T.: A review of metrics on permutations for search landscape analysis. *Comput. Oper. Res.* **34**, 3143–3153 (2007)
- Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Trans. Evol. Comput.* **7**, 117–132 (2003)