Supplementary Materials: Some Multi-Objective Local Search Algorithms Are Better than Others

Anonymous submission

Introduction

This supplementary document provides additional material for the main submission titled *"Some Multi-Objective Local Search Algorithms Are Better than Others"* to the 40th annual AAAI conference on artificial intelligence.

The rest of the document is structured as follows. The first section introduces the multi-objective combinatorial optimisation problems considered in this paper. The second and third sections give the experimental settings and additional experimental results for larger problem sizes. The fourth section introduces the procedure of investigating the distribution of the number of good neighbours. Finally, the fifth section gives the additional proofs regarding the two toy examples in the main paper.

Multi-Objective Combinatorial Problems

We consider four commonly used MOCOPs, the multi-objective 0/1 knapsack (Teghem 1994), travelling salesman problem (TSP) (Ribeiro et al. 2002), quadratic assignment problem (QAP) (Knowles and Corne 2003) and NK-landscapes (Aguirre and Tanaka 2004). Each problem was instantiated in three sizes (100, 200 and 500 decision variables).

Multi-Objective 0-1 Knapsack Problem (Knapsack). The multi-objective 0-1 knapsack problem (Teghem 1994) is a widely studied MOCOP. Given a set of D items $\mathbf{x} = (x_1, x_2, \dots, x_D) \in \{0,1\}^D$, the m-objective problem is defined as the following.

$$\begin{aligned} \max\ f_j(x) = \sum_{i=1}^{D} v_{ji} x_i, \quad j = 1, \dots, m \\ \text{s.t.} \sum_{i=1}^{D} w_i x_i \leq c \end{aligned} \tag{1}$$

Here, $v_{ji} \geq 0$ is the value of the item i in objective j, w_i is the item's weight, and $c = \frac{1}{2} \sum w_i$ is the capacity. Following (Li et al. 2024), both v_{ji} and w_i are sampled uniformly from $\{10, 11, \dots, 100\}$.

Multi-Objective Travelling Salesman Problem (TSP). The multi-objective TSP extends the classical TSP, with multiple costs between each pair of cities (Ribeiro et al. 2002), and aims to find the route minimising multiple travelling costs for visiting all the cities exactly once, returning to the start. Formally, given a network $L = (V, C)$, where $V = \{v_1, \dots, v_D\}$ is a set of D nodes and $C = \{C_j : j \in \{1, \dots, m\}\}$ is a set of m cost matrices between nodes $(C_j : V \times V)$, the problem is to find the Pareto optimal set of Hamiltonian cycles that minimise each of the m cost objectives.

Multi-Objective Quadratic Assignment Problem (QAP). The multi-objective QAP (Knowles and Corne 2003) models facility-location assignments with multiple flow types. Given m cost matrices $[C_{1,i,j}], \dots, [C_{m,i,j}]$ and a distance matrix $[L_{u,v}]$, a solution is a permutation $x = (x_1, ..., x_D)$ where x_i denotes the location of facility i. The problem is defined as the following.

$$\min\ f_k(x) = \sum_{i=1}^{D} \sum_{j=1}^{D} C_{k,i,j} L_{x_i, x_j}, \quad k = 1, \dots, m \tag{2}$$

Multi-Objective NK-Landscape. NK-landscapes (Aguirre and Tanaka 2004) are widely used due to their tunable ruggedness (Verel et al. 2013). Given a bit-string of length N and epistasis degree K (i.e., each variable is influenced by K other variables, collectively referred to as its locus), each objective f_j is defined as:

$$\begin{aligned} \max f_j(x) = \frac{1}{D} \sum_{i=1}^{D} c_{ij}(x_i, x_{k_{ij1}}, ..., x_{k_{ijK}}), \\ j = 1, \dots, m. \end{aligned} \tag{3}$$

Where c_{ij} represent the fitness contribution of the i-th variable, influenced by K other variables in its locus that can affect its contribution to the j-th objective. Each c_{ij} depends on the values of the i-variable and the variables in its locus, resulting in 2^{K+1} possible combinations of input combination and corresponding output values. Each output is randomly sampled from $(0, 1]$. Following (Aguirre and Tanaka 2007; Daolio et al. 2015), the K other variables of a variable's locus are drawn independently and uniformly at random for each i (variable) and j (objective), resembling a random epistasis pattern.

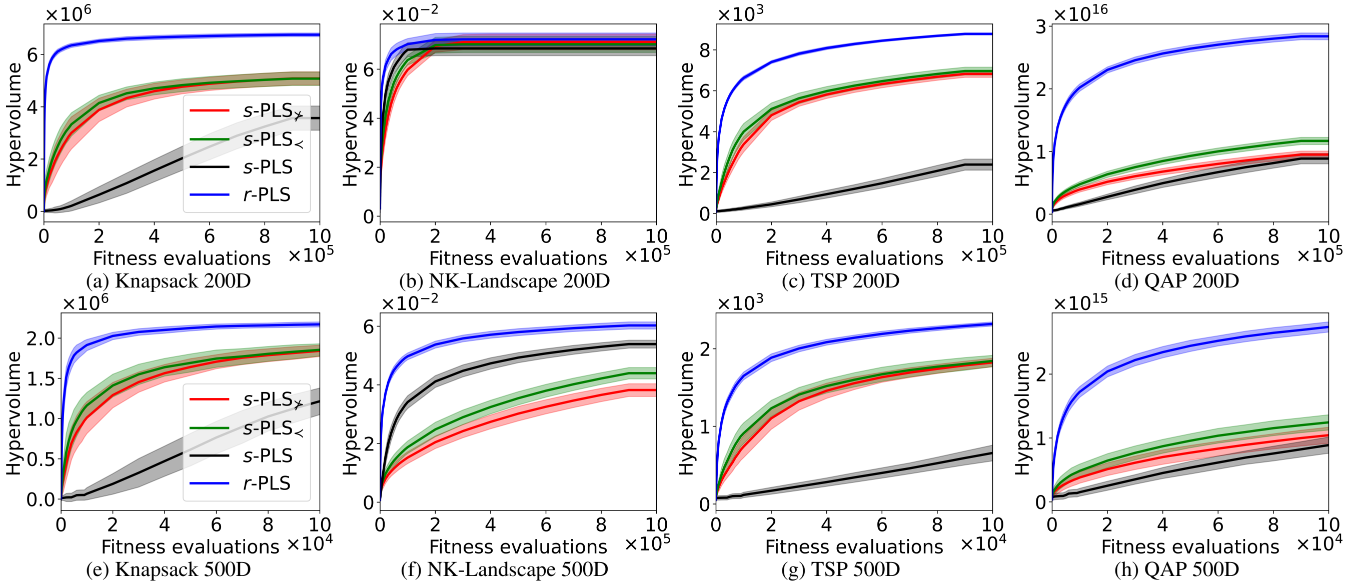

Figure 1: The hypervolume trajectory (the higher the better) of the considered s-PLS, s-PLS$_{\nprec}$, s-PLS$_{\prec}$ and r-PLS across 30 runs on the four MOCOPs with 200 variables (the upper panel) and with 500 variables (the lower panel). The bolded line and shaded area represent the mean and standard deviation of the hypervolume, respectively.

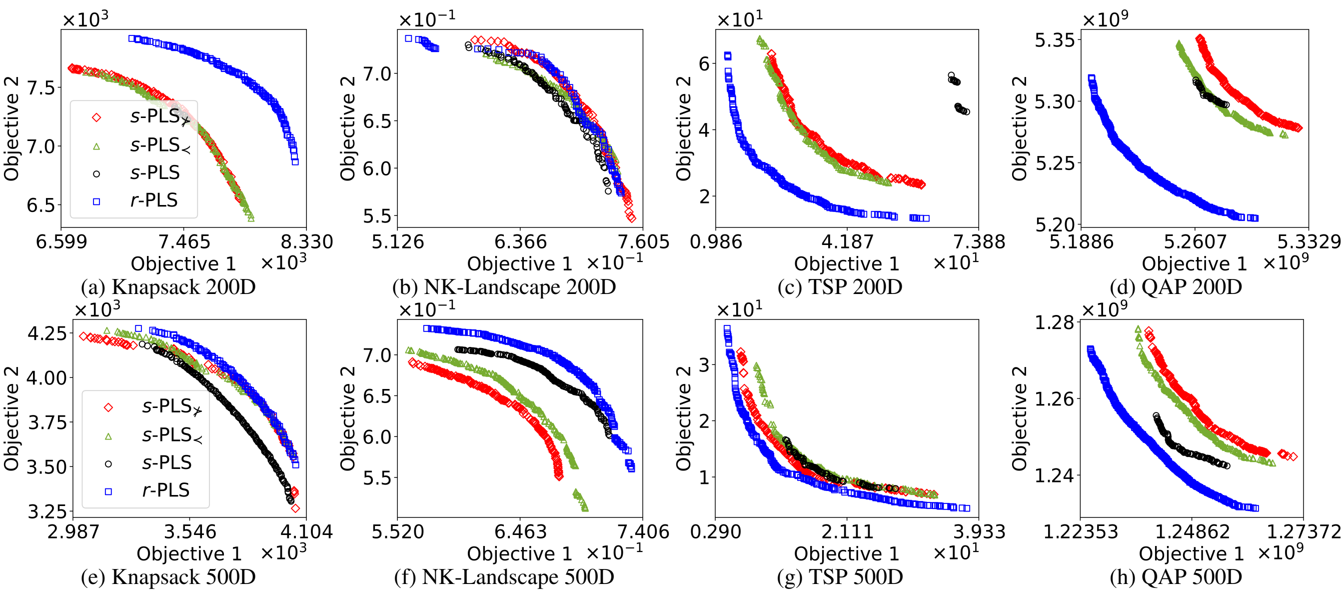

Figure 2: Non-dominated solutions obtained by the considered s-PLS, s-PLS$_{\nprec}$, s-PLS$_{\prec}$ and r-PLS in a typical run on the four MOCOPs with 200 variables (the upper panel) and with 500 variables (the lower panel), where the Knapsack and NK-Landscape are maximisation problems, and the TSP and QAP are minimisation problems.

Experimental Settings

In our experiments on the four MOCOPs, we consider four problem sizes (100, 200, and 500), each with a budget of 1,000,000 fitness evaluations. For illustration purposes, the results for problem size 100 presented in the main paper are based on only 100,000 fitness evaluations, as s-PLS terminates early for this size due to its stopping condition.

It is important to note that r-PLS and s-PLS use different stopping conditions: r-PLS runs until the evaluation budget is exhausted, whereas s-PLS may terminate earlier when there are no more unexplored solutions in the archive. To ensure fairness, all algorithms start from the same initial solution, and r-PLS is forcibly terminated when s-PLS terminates. Each instance is executed independently 30 times.

Neighbourhood operators are matched to the encoding of the variables for each problem: NK-landscape uses 1-bit flip (binary encoding), TSP uses 2-opt (order-based permutation encoding), and QAP uses 2-swap (position-based permutation encoding). The Knapsack problem requires a 2-bit flip neighbourhood, as 1-bit flip often fail to yield feasible or improved solutions near the constraint boundary.

We evaluate performance using the hypervolume (HV) indicator (Zitzler and Thiele 1999), with respect to a reference point estimated via random sampling. Specifically, we generate 100,000 random solutions from the decision space and only retain the non-dominated ones. Then, we define the reference point as the nadir point of these non-dominated sampled solutions, following the empirical methodology proposed in (Li et al. 2024). This is necessary because using the reference point determined by the non-dominated combined set of all generated solutions may easily cause the HV to be zero (Li, Chen, and Yao 2022; Li et al. 2024).

Additional Results

In this section, we present the additional experimental results on the four MOCOPs with the problem sizes of 200 and 500.

Figure 1 shows the average hypervolume (bolded line) and standard deviation (shaded area) of the basic *s*-PLS and *r*-PLS, as well as the two *s*-PLS variants across 30 independent runs on the four problems. The upper row corresponds to problems with 200 variables, and the lower row to those with 500 variables. Across all problem types, *r*-PLS consistently achieves higher hypervolume values than all *s*-PLS variants throughout the search on all problems. It also obtains better final performance in most cases – except for the NK-landscape with 200 variables, where all algorithms perform similarly.

Figure 2 gives the non-dominated solutions obtained by the four algorithms in a typical run on the four problems when the search ends. The layout mirrors that of Figure 1, with smaller problems on the top row and larger ones on the bottom. For the Knapsack and NK-Landscape problems (which are maximisation problems), better solutions lie toward the top-right corner; for TSP and QAP (minimisation problems), the desirable region is the bottom-left. As shown, *r*-PLS produces solutions with superior convergence and diversity compared to the three *s*-PLS algorithms. In constrast, the *s*-PLS algorithms, in many cases, are dominated by the *r*-PLS.

Together, these plots clearly demonstrate that *r*-PLS achieves better performance across both convergence and diversity, especially as problem size increases. These trends hold consistently across all four MOCOPs.

Distribution Fit At Each Collection Time

This section describes the details in investigating the distribution of the number of good neighbours of the solutions in the archive of *s*-PLS and *r*-PLS during the search. To find models that can describe the number of good neighbours, at each collection time, we consider several common discrete distributions as candidates, namely uniform, Poisson, geometric, binomial and Zipf. Additionally, we include the categorical distribution as a fallback model, since it can fit any empirical data regardless of underlying structure. Parameters of the distributions were estimated via maximum likelihood: Poisson's λ, geometric's p and binomial's p by sample mean; Zipf's exponent s via numerical minimisation of negative log-likelihood over $[1.01, 10]$; and categorical frequencies from empirical counts.

We evaluate the absolute fit quality using the χ^2 goodness-of-fit test at a significance level of $\alpha = 5\%$. A fit is considered acceptable if the test does not reject the null hypothesis that the observed data comes from the tested distribution. To compare the relative quality of fit between distributions, we rank all six fits using the Akaike's Information Criterion (AIC) (Akaike 1974). The degrees of freedom for each model are calculated as $n_{\text{bin}} - 1 - k$ where n_{bin} is the number of solutions that have at least one good neighbour and k is the number of free parameters of the distribution. A low AIC indicates a better goodness of fit while being less penalised by the model complexity (i.e., less free parameters). If the categorical distribution achieves the best AIC, this suggests the absence of a simple generative pattern in the data.

Figure 3 plots the goodness-of-fit of the five discrete distributions with respect to the numbers of good neighbours of solutions in the archive during the search process of *s*-PLS and *r*-PLS on the four MOCOPs. Each horizontal band corresponds to a candidate distribution, and at each sampled evaluation, two coloured ticks, black and blue, respectively indicate a good fit for *s*-PLS and *r*-PLS at $\alpha = 0.05$, namely, the χ^2 test does not reject the distribution. As can be seen in the figure, across nearly all problem instances and time-scenarios, the geometric distribution more frequently passed the χ^2 test significantly than others, indicating that the number of good neighbours is more consistent with a geometric distribution. In contrast, The uniform distribution is only fitted and more fitted than others for PLS at the first to four $|\mathcal{N}|$ searches, as it is basically telling the quality of neighbouring solutions or the initial solution, but is not fitted in other cases. The Poisson distribution only fits in the later stages of the search, where the geometric distribution also fits, as well. The binomial usually has the worst AIC on permutation problems (e.g., TSP), and a second worst on binary problems (e.g., knapsack). Zipf's distribution cannot capture the large spike at zero (indicating no good neighbours at the later stage of the search).

As for the relative fit, if multiple distributions fit at the same time, the geometric distribution also achieved the lowest or the second-lowest AIC by margins often exceeding ten to the next (i.e. "strong" over second-best). On the other hand, Zipf's AIC is often the worse.

Proofs of The Two Examples

Proposition 1 (Half good and half bad solutions)**.** *Let an archive of n solutions contains exactly $n/2$ solutions have no good neighbours ($G = 0$), and another $n/2$ solutions with all neighbours are good ($G = |\mathcal{N}|$). Then, given $|\mathcal{N}| \geq 2$,* r-*PLS has a smaller expected runtime than* s-*PLS in finding the next new good solution.*

Proof. For *r*-PLS, with probability $\frac{1}{2}$, the selected solution is full of good neighbours, and a uniformly drawn neighbour

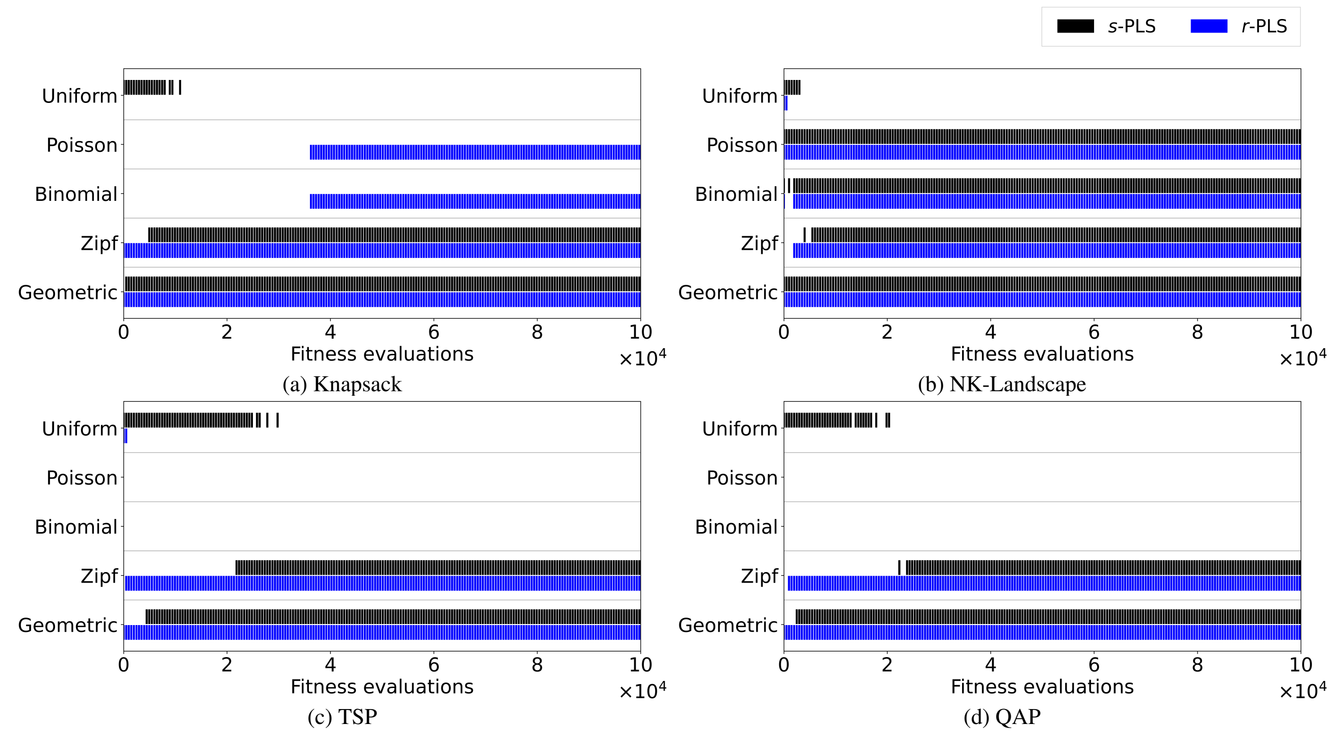

Figure 3: Goodness-of-fit of the distributions with respect to the number of good neighbours of solutions in the archive during the search process of s-PLS (black) and r-PLS (blue) on the (a) Knapsack (100 items), (b) NK-Landscape (N=100, K=10), (c) TSP (100 cities) and (d) QAP (100 factories). A coloured tick in a row indicates that the corresponding algorithm's data at that point was not rejected under the model.

is automatically good. Thus,

$$p_{\text{succ}} = \tfrac{1}{2}, \qquad \mathbb{E}[T_{r\text{-PLS}}] = \frac{1}{p_{\text{succ}}} = \frac{1}{1/2} = 2. \tag{4}$$

As for s-PLS, it wastes $|\mathcal{N}|$ evaluations for every "bad" solution encountered at the beginning until it reaches a "good" solution. Hence, we need to first estimate the timing that s-PLS finds a "good" solution. s-PLS typically scan through each archive solution one-by-one as it remember the explored solution. Therefore, the above estimation resembles the process of finding the first index of a good neighbour during scan. By lemma 1, the first solution with all good neighbours appears at index

$$\mathbb{E}[J] = \frac{n+1}{(n/2)+1} = \frac{2(n+1)}{n+2}.$$

Since each preceding "bad" solution consumes $|\mathcal{N}|$ evaluations and the first solution with all good neighbours consumes exactly 1, the expected runtime of s-PLS is the following:

$$\begin{aligned}\mathbb{E}[T_{s\text{-PLS}}] &= \big(\mathbb{E}[J]-1\big)|\mathcal{N}|+1\\ &= \left(\frac{2(n+1)}{n+2}-1\right)|\mathcal{N}|+1 = \frac{n|\mathcal{N}|}{n+2}+1.\end{aligned} \tag{5}$$

Since $|\mathcal{N}| \geq 2$, it follows that $\mathbb{E}[T_{s\text{-PLS}}] > \mathbb{E}[T_{r\text{-PLS}}]$. □

Proposition 2 (Half neighbours are good for all solutions)**.** *Let an archive of n solutions and each solution has exactly half good neighbours ($G = |\mathcal{N}|/2$). Then, given $|\mathcal{N}| \geq 2$,* s-*PLS has a smaller expected runtime than* r-*PLS in finding the next new good solution.*

Proof. For r-PLS, with probability $\frac{1}{2}$, it picks a good neighbour from a randomly selected solution. Thus,

$$p_{\text{succ}} = \tfrac{1}{2}, \qquad \mathbb{E}[T_{r\text{-PLS}}] = \frac{1}{p_{\text{succ}}} = \frac{1}{1/2} = 2. \tag{6}$$

As for s-PLS, we only need to consider the neighbourhood of one solution since all solutions have half of their neighbours being good. By lemma 1, the first good neighbour appears at the index of the neighbourhood at $E[J] = \frac{|\mathcal{N}|+1}{(|\mathcal{N}|/2)+1}$, resembles the runtime of s-PLS:

$$\mathbb{E}[T_{s\text{-PLS}}] = E[J] = \frac{|\mathcal{N}|+1}{(|\mathcal{N}|/2)+1} < 2 = \mathbb{E}[T_{r\text{-PLS}}].$$

□

References

Aguirre, H.; and Tanaka, K. 2007. Working principles, behavior, and performance of MOEAs on MNK-landscapes. *European Journal of Operational Research*, 181(3): 1670–1690.

Aguirre, H. E.; and Tanaka, K. 2004. Effects of elitism and population climbing on multiobjective MNK-landscapes. In *Congress on Evolutionary Computation*, volume 1, 449–456.

Akaike, H. 1974. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6): 716–723.

Daolio, F.; Liefooghe, A.; Verel, S.; Aguirre, H.; and Tanaka, K. 2015. Global vs local search on multi-objective NK-landscapes: contrasting the impact of problem features. In *Genetic and Evolutionary Computation Conference*, 369–376.

Knowles, J.; and Corne, D. 2003. Instance generators and test suites for the multiobjective quadratic assignment problem. In *Evolutionary Multi-Criterion Optimization*, 295–310.

Li, M.; Chen, T.; and Yao, X. 2022. How to evaluate solutions in Pareto-based search-based software engineering? A critical review and methodological guidance. *IEEE Transactions on Software Engineering*, 48(5): 1771–1799.

Li, M.; Han, X.; Chu, X.; and Liang, Z. 2024. Empirical Comparison between MOEAs and Local Search on Multi-Objective Combinatorial Optimisation Problems. In *Genetic and Evolutionary Computation Conference*, 547–556. ACM.

Ribeiro, C. C.; Hansen, P.; Borges, P. C.; and Hansen, M. P. 2002. A study of global convexity for a multiple objective travelling salesman problem. *Essays and Surveys in Metaheuristics*, 129–150.

Teghem, J. 1994. Multi-objective combinatorial optimization problems: A survey. *Journal of Multi-Criteria Decision Analysis*.

Verel, S.; Liefooghe, A.; Jourdan, L.; and Dhaenens, C. 2013. On the structure of multiobjective combinatorial search space: MNK-landscapes with correlated objectives. *European Journal of Operational Research*, 227(2): 331–342.

Zitzler, E.; and Thiele, L. 1999. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4): 257–271.