IMS-LOMONAS: Pareto Local Search for Multi-objective Neural Architecture Search

Quan Minh Phan, Ngoc Hoang Luong
University of Information Technology, Ho Chi Minh City, Vietnam
Vietnam National University, Ho Chi Minh City, Vietnam
quanphm@uit.edu.vn, hoangln@uit.edu.vn

I. Algorithm Description

The algorithm that we submit for the competition namely IMS-LOMONAS [1], which is the parameter-less version of LOMONAS [2] by utilizing the interleaved multi-start scheme.

The LOMONAS [2] algorithm (in short for **Lo**cal search for **M**ulti-**O**bjective **N**eural **A**rchitecture **S**earch) is an iterated Pareto local search (PLS) that repeats the PLS till it satisfies the stopping conditions (e.g., reaching the maximum number of evaluations). In contrast to previous PLS algorithms that only perform the local improvement on non-dominated solutions at each iteration, LOMONAS first performs the Non-dominated Sorting mechanism [3] to assign the ranks of solutions and then performs the local improvement on solutions that are on k lowest-rank fronts. Moreover, instead of improving all solutions on the selected fronts, LOMONAS only improves the *knee* and *extreme* solutions for the sake of efficiency. The procedure of LOMONAS is presented in Algorithm 1.

In LOMONAS, the knee solutions on the fronts are identified via the angle-based approach [4]. Specifically, if a solution $\boldsymbol{x}$ and its two nearest neighbors (in the objective space) together form an angle that is larger than a pre-defined angle value $\measuredangle$, the solution $\boldsymbol{x}$ can be considered as a knee one [4]. Since this approach only effectively works with 2-dimensional objective space (i.e., bi-objective problem), we have extended it to deal with many-objective NAS problems. When we deploy LOMONAS on many-objective NAS problems (i.e., the number of objectives is greater than 2), we first re-project the current high-dimensional objective space into many 2-dimensional objective spaces. The knee solutions on each subspace are detected by employing the angle-based approach and are considered for the local improvement step.

Choosing a specific effective value for the number of selected fronts k in LOMONAS is challenging when empirical results indicate that a low k-value could bring better results to LOMONAS than high k-values in certain problems and *vice versa*. Moreover, hyperparameter tuning in NAS is typically impractical due to the expensive computational cost of a trial run. We thus deploy the interleaved multi-start scheme (IMS) on LOMONAS, resulting in IMS-LOMONAS, to remove the manual setting of the hyperparameter k in LOMONAS. The core idea of combining LOMONAS with IMS is executing various versions of LOMONAS with gradually increasing the values of k. Initializing a new LOMONAS instance or executing the previous LOMONAS instances is scheduled following a counter b. Fig 1 illustrates the work schedule of IMS-LOMONAS with the counter $b = 4$. During the search of IMS-LOMONAS, we additionally eliminate the *inefficient* LOMONAS instances to increase the efficiency. The pseudocode of IMS-LOMONAS is exhibited in Algorithm 2. For each LOMONAS instance, each time of performing local improvement process corresponds to execute lines $8 - 22$ in Algorithm 1.

Algorithm 1: LOMONAS

Input: The number of selected fronts k; The search space of architectures Ω_{arch}; The angle value for selecting knee solutions $\measuredangle$

Output: The elitist archive $\mathcal{A}$.

```
1  A ← ElitistArchive()
2  x^start ← Sample(Ω_arch)
3  while ¬TerminationCriteriaSatisfied do
4      Evaluate(x^start);   A.update(x^start)
5      S ← {x^start};   Q ← {x^start}
6      while True do
7          // Local Improvement
8          N ← GetUnvisitedNeighbors(Q)
9          if N = ∅ then
10             for i ∈ {1, 2, ..., k − 1} do
11                 Q ← GetKneeAndExtremeSolutions(S, ∡, i)
12                 N ← GetUnvisitedNeighbors(Q)
13                 if N = ∅ then break
14             if N = ∅ then
15                 x ← Sample(A)
16                 N_x ← GetAllNeighbors(x)
17                 x^start ← Sample(N_x)
18                 break
19         for x ∈ N do
20             Evaluate(x);   A.update(x)
21         S ← GetK-LowestRankFronts(S ∪ N, k)
22         Q ← GetKneeAndExtremeSolutions(S, ∡, 0)
23 return A
```

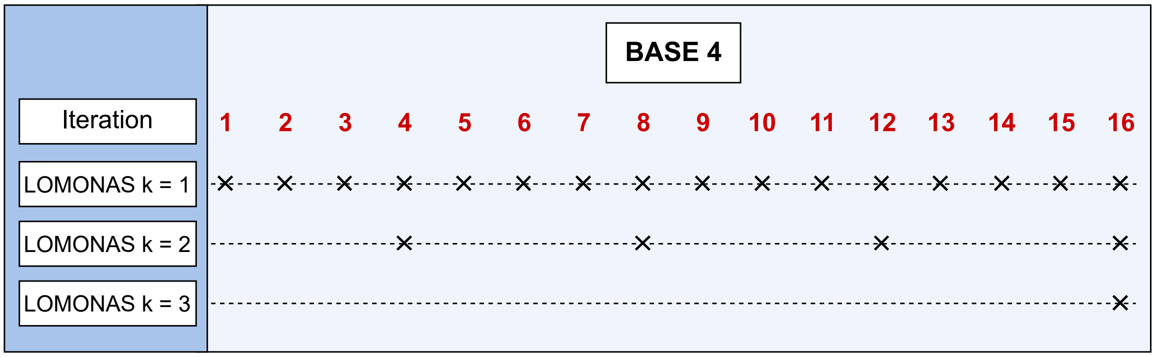

Fig. 1. Example of adapting the interleaved multi-start scheme with the counter $b = 4$ for getting rid of setting the number of selected fronts k in LOMONAS.

Algorithm 2: IMS-LOMONAS

Input: Counter of base b; The search space of architectures Ω_{arch}; The angle value for selecting knee solutions $\measuredangle$

Output: Elitist archive $\mathcal{A}_{overall}$.

$\mathcal{A}_{overall} \leftarrow$ ElitistArchive()
$n\_iter \leftarrow 0$; $t \leftarrow 1$
$k \leftarrow 1$
$\mathcal{C} \leftarrow [\ ]$
$instance \leftarrow$ Initialize-LOMONAS(k, Ω_{arch}, $\measuredangle$)
$\mathcal{C}.add(instance)$
while $\neg$TerminationCriteriaSatisfied **do**
  $n\_iter \leftarrow n\_iter + 1$
  if $n\_iter = b^t$ **then**
    $k \leftarrow k + 1$
    $instance \leftarrow$ Initialize-LOMONAS(k, Ω_{arch}, $\measuredangle$)
    $\mathcal{C}.add(instance)$
    $t \leftarrow t + 1$
  for $i \in \{0, 1, \ldots, |\mathcal{C}| - 1\}$ **do**
    if $(n\_iter \bmod b^i) = 0$ **then**
      while $\mathcal{C}[i].n\_eval < \mathcal{C}[i-1].n\_eval$ / $log_2(b)$ **do**
        $\mathcal{C}[i].run\_one\_local\_improvement()$
  $\mathcal{C} \leftarrow$TerminateDominatedInstances($\mathcal{C}$)
return $\mathcal{A}_{overall}$

II. Experiments

All experiments are conducted on **Python 3.9** with the latest version of **EvoXBench** (1.0.5) and **pymoo** (0.6.1) packages. We do not set the population size following settings in the competition because the IMS-LOMONAS does not have the population. The hyperpameters of IMS-LOMONAS for all test problems is set as follows:

- Counter of base $b = 8$
- Angle value for selecting knee solutions $\measuredangle = 210°$

We run the IMS-LOMONAS 31 independent times for each problem. The maximum number of evaluations at each trial is set to 10,000 and the hypervolume (HV) indicator is used to evaluate the performance of algorithm following the requirements. The results of IMS-LOMONAS on 15 test instances are presented in Table I.

TABLE I
Mean and standard deviation (in brackets) of the hypervolume indicator of IMS-LOMONAS on the CEC'2024 test instances.

Problem	D	M	IMS-LOMONAS
CitySeg/MOP1	32	2	0.9001 (0.0018)
CitySeg/MOP2	32	3	0.7979 (0.0016)
CitySeg/MOP3	32	3	0.8223 (0.0016)
CitySeg/MOP4	32	4	0.6986 (0.0004)
CitySeg/MOP5	32	5	0.6564 (0.0004)
CitySeg/MOP6	32	2	0.7678 (0.0021)
CitySeg/MOP7	32	3	0.7277 (0.0029)
CitySeg/MOP8	32	3	0.7282 (0.0031)
CitySeg/MOP9	32	4	0.5760 (0.0005)
CitySeg/MOP10	32	5	0.5465 (0.0005)
CitySeg/MOP11	32	3	0.6904 (0.0006)
CitySeg/MOP12	32	5	0.4760 (0.0004)
CitySeg/MOP13	32	6	0.4244 (0.0002)
CitySeg/MOP14	32	6	0.4366 (0.0003)
CitySeg/MOP15	32	7	0.3993 (0.0002)

References

[1] Q. M. Phan and N. H. Luong, "Parameter-less Pareto local search for multi-objective neural architecture search with the Interleaved Multi-start Scheme," *Swarm Evol. Comput.*, 2024.
[2] ——, "Pareto Local Search is Competitive with Evolutionary Algorithms for Multi-Objective Neural Architecture Search," in *GECCO*, 2023.
[3] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, 2002.
[4] J. Branke, K. Deb, H. Dierolf, and M. Osswald, "Finding Knees in Multi-objective Optimization," in *PPSN*, 2004.