KLP Not Always Efficient

Sanyou Zeng[1,2], Lixin Ding[3], Shuzhen Yao[2], and Lishan Kang[3]

[1] Dept. of Computer Science, Zhuzhou Institute of Technology, 412008 Zhuzhou, Hunan, P.R.China
{sanyou-zeng}@263.net
[2] Dept. of Computer Science, China University of GeoSciences, 430074 Wuhan, Hubei, P.R.China
[3] State Key Laboratory of Software Engineering, Wuhan University 430072 Wuhan, Hubei, P.R.China

Abstract. The size of the nondominated set of a vector set is greatly dependent on the size of the original vector set N and the dimension of the vector M. Theoretical analysis shows that when $M = O(\log N)$ the original set has big nondominated set which may be the original set itself, and in the case $M = O(\log N)$, a classical algorithm (KLP) for finding nondominated set has complexity of KLP higher than N^2. Experiment verifies the analysis result as well. Therefore, we should avoid employing KLP when $M = O(\log N)$.
Keywords: nondominated set, partial ordered relation, multi-objective optimization.

1 Introduction

To solve a multi-objective problem is to find the set of Pareto-optimal solutions, mathematically, the nondominated set of the search space. Classical optimization methods can at best find one solution in one simulation run, while evolutionary algorithms (EAs) can find multiple optimal solutions in one single simulation run due to their population approach. Thus EAs are ideal candidates for solving multi-objective optimization problems. Representative evolutionary techniques include vector evaluated genetic algorithm (VEGA)[Schaffer1985], Pareto-based ranking procedure (FFGA) [FonsecaFleming1993], niched Pareto genetic algorithm (NPGA) [Horn1994], Non-dominated Sorting Genetic Algorithm (NSGA) [SrinivasDeb1994], strength Pareto evolutionary algorithm (SPEA) [Zitzler1999], Generalised Regression GA (GRGA) [Tiwari2002]. Recently, there are some new approaches or modified versions of proposed methods, for instance, NSGAII [Deb2001a], SPEA2 [Zitzler2001], rMOGAxs [PurshouseFleming2001] , Pareto-Archived Evolution Strategy (PAES) [Knowles2000] and so on. All multi-objective evolutionary algorithms (MOEA) can't avoid searching a nondominated set, and the pitfall of MOEAs is in some way time consuming. Therefore, it is important to take a fast procedure for finding the nondominated set. Deb argues KLP algorithm [Kung1975] as an effective algorithm in his new book [Deb2001b], and it seems that KLP is a fast algorithm with computational complexity lower than N^2.

In this paper, the dependence of the size of the nondominated set of a vector set on the size of the original vector set N and the dimension of the vector M is analyzed. Theoretical analysis shows that when $M = O(\log N)$ the original set has big nondominated set which may be the original set itself, and in the case $M = O(\log N)$, a classical algorithm (KLP) for finding nondominated set has complexity of KLP higher than N^2. Experiment verifies the analysis result as well. Therefore, we should avoid employing KLP when $M = O(\log N)$.

In the remainder of the paper, we briefly mention nondominace, nondominated set and two existed algorithms for finding nondominated set in section 2. Then in Section 3, the dependency of the size of the nondominated set for $\mathbf{X}$ on N and M is analyzed. Thereafter, we give a remark on the computational complexity of KLP in Section 4. The next section presents numerical experiments. Finally, we summarize the conclusions of this paper.

2 Non-dominated Set

Let $\mathbf{X}$ be a vector set $\mathbf{X} = \{\mathbf{x}^{(i)} | i = 1, 2, ..., N\}$ where $\mathbf{x}^{(i)} = (x_1^{(i)}, x_2^{(i)}, ..., x_M^{(i)})$ is vector. $\mathbf{X}$ is regarded as $N \times M$ matrix $\mathbf{X} = [x_j^{(i)}]_{N\times M}$. The size of the $\mathbf{X}$ is N and the dimension of vector is M. Applied to multiobjective evolutionary algorithm, $\mathbf{X}$ is population, where M is the number of objectives and N the number of individuals $\mathbf{x}^{(i)}$ in the population.

Definition 1. *For any two vectors $\mathbf{x}^{(i_1)}, \mathbf{x}^{(i_2)} \in \mathbf{X}$,*

$$\begin{aligned}
\mathbf{x}^{(i_1)} = \mathbf{x}^{(i_2)} &\iff \forall j \in 1, 2, ..., M : x_j^{(i_1)} = x_j^{(i_2)} \\
\mathbf{x}^{(i_1)} \preceq \mathbf{x}^{(i_2)} &\iff \forall j \in 1, 2, ..., M : x_j^{(i_1)} \leq x_j^{(i_2)} \\
\mathbf{x}^{(i_1)} \prec \mathbf{x}^{(i_2)} &\iff \begin{cases} \mathbf{x}^{(i_1)} \preceq \mathbf{x}^{(i_2)} \\ \mathbf{x}^{(i_1)} \neq \mathbf{x}^{(i_2)} \end{cases} \\
\mathbf{x}^{(i_1)} \sim \mathbf{x}^{(i_2)} &\iff \begin{cases} \exists j_0, x_{j_0}^{(i_1)} < x_{j_0}^{(i_2)} \\ \exists j_1, x_{j_1}^{(i_2)} < x_{j_1}^{(i_1)} \end{cases}
\end{aligned} \tag{1}$$

The relation "$\preceq$" on $\mathbf{X}$ is a partial order relation, and "$\prec$" a strict partial order relation. $\mathbf{X}$ with relation "$\prec$" is a strict partial order set which we denote by $(\mathbf{X}, \prec)$.

Definition 2. *A member $\mathbf{x}' \in \mathbf{X}$ is said to be a **nondominated member** of $(\mathbf{X}, \prec)$ iff*

$$\nexists \mathbf{x} \in \mathbf{X} : \mathbf{x} \prec \mathbf{x}' \tag{2}$$

and

$$\mathbf{M}(\mathbf{X}, \prec) = \{\mathbf{x} | \mathbf{x} \in \mathbf{X} \text{ is a nondominated member}\} \tag{3}$$

*is called the **nondominated set** of $(\mathbf{X}, \prec)$.*

In the following, two existed algorithms for finding nondominated set are introduced.

The first approach employs a good bookkeeping technique which continuously updates check states of members. Let **CU** denote this approach.

Algorithm 1 *Identifying the Non-dominated Set:CU*
Let $\mathbf{X}'$ *store the nondominated set of* $\mathbf{X}$*,* $|\mathbf{X}'|$ *be the cardinality of* $\mathbf{X}'$*.*

$\mathbf{X}' = \{\mathbf{x}^{(1)}\};$
$for(i = 2; i \leq N; i++)\{$
  $//Detemine\ if\ \mathbf{x}^{(i)}\ currently\ nondominated.$
  $bool\ i\_nondominated = true;$
  $for(t = 1; t \leq |\mathbf{X}'|; t++)\{$
    $if(\mathbf{x}'^{(t)} \prec \mathbf{x}^{(i)})\{//\mathbf{x}'^{(t)}\ is\ the\ tth\ member\ in\ \mathbf{X}'$
      $i\_nondominated = false;\ break;$
    $\}$
    $if(\mathbf{x}^{(i)} \prec \mathbf{x}'^{(t)})Delete\ \mathbf{x}'^{(t)}\ from\ \mathbf{X}';$
  $\}$
  $if(i\_nondominated == true)Insert\ \mathbf{x}^{(i)}\ into\ \mathbf{X}';$
$\}$
$Output\ \mathbf{X}';$

The algorithm requires $O(N^2)$ comparisons for nondominace and each comparison needs an average of numerical comparisons about 3, and therefore, the average computational complexity is $O(N^2)$. We should note that the most computations required is $O(MN^2)$.

The second approach was proposed by Kung, H. T., Luccio, F. and Preparata, F. P. [Kung1975], it is said to be an efficient one. Let us denote it by KLP.

Algorithm 2 *Identifying the Non-dominated Set:KLP*

1. *Sort the set* $\mathbf{X}$ *according to the descending order of importance of the first column component.*
2. *Nondominance(*$\mathbf{X}$*): If* $|\mathbf{X}| = 1$*, then return* $\mathbf{X}$ *as the output of Nondominance(* $\mathbf{X}$ *).*
   Otherwise, $\mathbf{T}$ = *Nondominance (*$\mathbf{X}^{(1)} - \mathbf{X}^{(\frac{|\mathbf{X}|}{2})}$*) and* $\mathbf{B}$ = *Nondominance (*$\mathbf{X}^{(\frac{|\mathbf{X}|}{2}+1)} - \mathbf{X}^{(|\mathbf{X}|)}$*), where* $\mathbf{X}^{(1)} - \mathbf{X}^{(\frac{|\mathbf{X}|}{2})}$ *is the top half of* $\mathbf{X}$*,* $\mathbf{X}^{(\frac{|\mathbf{X}|}{2}+1)} - \mathbf{X}^{(|\mathbf{X}|)}$ *the bottom half. For any member* $\mathbf{x}^{(i)} \in \mathbf{B}$*, if* $\mathbf{x}^{(i)}$ *is not dominated all members in* $\mathbf{T}$*, create a merged set* $\mathbf{M} = \mathbf{T} \bigcup \{\mathbf{x}^{(i)}\}$*. Return* $\mathbf{M}$ *as the output of the output of Nondominance(*$\mathbf{X}$*).*

The complexity of this approach is $O(N(\log N)^{M-2})$ for $M \geq 4$ and $O(N(\log N))$ for $M = 2, 3$.

For more relative information, see the literature [Deb2001b].

3 Dependency of the Size of the Nondominated Set for $\mathbf{X}$ on N, M

In $\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(N)}\} = [x_j^{(i)}]_{N\times M}$, suppose $x_j^{(i)}$ is a random variable evenly distributed over real number set and any two such variables are independent. We have

$$P[x_j^{(i)} < x_l^{(k)}] = \tfrac{1}{2}, P[x_l^{(k)} < x_j^{(i)}] = \tfrac{1}{2}, P[x_j^{(i)} = x_l^{(k)}] = 0$$

where $i \neq k$, $j \neq l$.

We know that the size of the nondominated set for $\mathbf{X}$ is greatly dependent on both N and M. Given N, let M increase from 1, then the size of nondominated set will increase bigger and bigger. There exists a threshold for M where $\mathbf{X}$ has a big nondominated set which may be $\mathbf{X}$ itself. Before M arrives at the threshold, the nondominated set is smaller and almost never the $\mathbf{X}$ itself. After M exceeds the threshold, the nondominated set is almost the $\mathbf{X}$ itself. In other words, before M arrives at the threshold, the probability of the event that the nondominated set of $\mathbf{X}$ is $\mathbf{X}$ itself (denoted as E_N) is 0. After M exceeds the threshold, the probability of the event E_N is $\mathbf{X}$ itself is 1. When M stays at the threshold, the probability of the event E_N is between 0 and 1. For quantizing the threshold, the probability of the event E_N should be calculated. We first consider the event $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(k)}$ are nondominated each other, that is, $\mathbf{x}^{(i)} \sim \mathbf{x}^{(k)}$ (cf. Definition 1), and denote it by $E_{i,j} = \{\mathbf{x}^{(i)} \sim \mathbf{x}^{(k)}\}$ where $\mathbf{x}^{(i)} = (x_1^{(i)}, x_2^{(i)}, ..., x_M^{(i)})$ and $\mathbf{x}^{(k)} = (x_1^{(k)}, x_2^{(k)}, ..., x_M^{(k)})$. It is easy to compute

$$P[E_{i,k}] = 1 - 2(\tfrac{1}{2})^M = 1 - (\tfrac{1}{2})^{M-1} \tag{4}$$

Then we have the following theorem

Theorem 1. *Denote event that* $\mathbf{X}$ *has itself as nondominated set* E_N*, its probability is*

$$P[E_N] = \left(1 - (\tfrac{1}{2})^{M-1}\right)^{\frac{N(N-1)}{2}} \tag{5}$$

Proof If $N = 2$, the formula (5) is obviously true. We now assume the formula (5) is true for $N = n$, that is,

$$P[E_n] = \left(1 - (\tfrac{1}{2})^{M-1}\right)^{\frac{n(n-1)}{2}}$$

and prove it is true when N replaced by $n+1$. For $N = n+1$, and $\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(n)}, \mathbf{x}^{(n+1)}\}$, the following is true

$$E_{n+1} = E_n \bigcap E_{1,n+1} \bigcap E_{2,n+1} \bigcap ... \bigcap E_{n,n+1}$$

From the creation of $\mathbf{X}$, we have that E_n, $E_{1,n+1}$, $E_{2,n+1}$, ..., $E_{n,n+1}$ are independent with each other, then, it follows

$$\begin{aligned} P[E_{n+1}] &= P[E_n] \times P[E_{1,n+1}] \times ... \times P[E_{n,n+1}] \\ &= \left(1 - (\tfrac{1}{2})^{M-1}\right)^{\frac{n(n-1)}{2}} \\ &\quad \times (1 - (\tfrac{1}{2})^{M-1}) \times ... \times (1 - (\tfrac{1}{2})^{M-1}) \\ &= \left(1 - (\tfrac{1}{2})^{M-1}\right)^{\frac{(n+1)n}{2}} \end{aligned}$$

This is the theorem with N replaced by $n+1$, and the theorem holds by induction. $\sharp$

Given N and let M vary. When M stays at the threshold, the probability of event E_N is between 0 and 1. Given such a probability, we have the following equation from equation (5),

$$M = 1 + \log_2\left(\frac{1}{1-P^{\frac{2}{N(N-1)}}}\right) \tag{6}$$

Since

$$\lim_{x\to 0}\frac{\frac{1}{1-P^x}}{\frac{1}{x}} = \lim_{x\to 0}\frac{x}{1-P^x} = \lim_{x\to 0}\frac{1}{-P^x\log P} = \frac{1}{\log P^{-1}} \tag{7}$$

Therefore

$$\frac{1}{1-P^x} = O\left(\frac{1}{x\log P^{-1}}\right) \tag{8}$$

Equation (8) with x replaced by $\frac{2}{N(N-1)}$, we have

$$\frac{1}{1-P^{\frac{2}{N(N-1)}}} = O\left(\frac{N(N-1)}{2\log P^{-1}}\right) \tag{9}$$

By equation (9) and equation (6), it follows that

$$\begin{aligned} M &= 1 + O(\log(\tfrac{N(N-1)}{2\log P^{-1}})) \\ &= O(\log N) - O(\log\log P^{-1}) \\ &= O(\log N) \end{aligned} \tag{10}$$

That is, the threshold is $M = O(\log N)$. When $M = O(\log N)$, $\mathbf{X}$ has a big nondominated set which may be $\mathbf{X}$ itself.

4 Remark on the Computation Complexity of KLP

4.1 Misunderstanding the Computational Complexity of KLP

Some people misunderstand that KLP has a computational complexity lower than $O(N^2)$. The cause is that they regard M as constant while N as variable. Therefore, it follows that

$$\begin{aligned} &\lim_{N\to\infty}\frac{N(\log N)^{M-2}}{N^2} = \lim_{N\to\infty}\frac{(\log N)^{M-2}}{N} \\ &= \lim_{N\to\infty}\frac{(M-2)(\log N)^{M-3}}{N} = \ldots = \lim_{N\to\infty}\frac{(M-2)!}{N} = 0 \end{aligned} \tag{11}$$

that is, the computational complexity of KLP $S_{KLP} = o(N^2)$ when M is constant. However, when N,M both variable. the above result can not follow.

4.2 Disadvantage of KLP

When M stays at the threshold, that is $M = O(\log N)$, the computational complexity of KLP will be higher than N^2. In fact, for $M = \log N$, it follows that

$$\begin{aligned} &\lim_{N\to\infty} \frac{N^2}{N(\log N)^{M-2}} = \lim_{N\to\infty} \frac{N}{(\log N)^{M-2}} \\ &= \lim_{N\to\infty} \frac{N}{(\log N)^{\log N-2}} = \lim_{N\to\infty} \frac{e^{\log N}}{e^{(\log N-2)\log\log N}} = 0 \end{aligned} \tag{12}$$

In this case, KLP will be slower than CU. As a summary, the computational complexity of KLP is lower than N^2 for M being a constant, while for $M \geq O(\log N)$, it is higher than N^2.

5 Numerical Experiments

Experiment was performed on a computer with Pentium 4/1.80GHz CPU and 256M memory. Table 1, 2, 3, 4 give the running time comparison of KLP with CU for finding nondominated set, and the results show that when M is small(or original set has small nondominated set), CU is slower than KLP, when M is relatively larger(or original set has a larger nondominated set), CU is faster than KLP.

Table 1. The time comparison of KLP with CU where $N = 1000$, M different. Q is the size of nondominated set.

$N \times M$	Running Time(Second)		Q
	CU	KLP	
1000×3	0.016	0.015	26
1000×5	0.032	0.032	172
1000×10	0.125	0.156	732
1000×20	0.171	0.180	999
1000×30	0.172	0.234	1000

6 Conclusion

The dependency of the size of the nondominated set of a vector set on the size of the original vector set N and the dimension of the vector M is theoretically analyzed. Theoretical and experimental results show that when M is small(or original set has small nondominated set), CU is slower than KLP, when M is relatively larger(or original set has a larger nondominated set), CU is faster than KLP.

Table 2. The time comparison of KLP with CU where $N = 5000$, M different. Q is the size of nondominated set.

$N \times M$	Running Time(Second)		Q
	CU	KLP	
5000×3	0.078	0.078	45
5000×5	0.344	0.203	317
5000×10	2.859	2.797	2962
5000×20	5.968	5.718	4978
5000×30	6.009	6.641	5000

Table 3. The time comparison of KLP with CU where $N = 10000$, M different. Q is the size of nondominated set.

$N \times M$	Running Time(Second)		Q
	CU	KLP	
10000×3	0.140	0.172	49
10000×5	0.532	0.391	278
10000×10	10.062	10.781	5091
10000×20	23.750	28.521	9930
10000×30	25.927	30.985	10000

Table 4. The time comparison of KLP with CU where $N = 20000$, M different. Q is the size of nondominated set.

$N \times M$	Running Time(Second)		Q
	CU	KLP	
20000×3	1.2820	0.344	68
20000×5	2.4530	0.907	513
20000×10	34.8910	36.500	8321
20000×20	96.844	129.086	19756
20000×30	105.021	137.172	20000

Acknowledgment This work was supported by National Natural Science Foundation of China (No.s: 60204001, 70071042, 60073043, 60133010), A Project Supported by Scientific Research Fund of Hunan Provincial Education Department(No. 02C640), and Youth Chengguang Project of Science and Technology of Wuhan City(No.:20025001002).

References

[Schaffer1985] Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In J. J. Grefenstette (Ed.),*Proceedings of an International Conference on Genetic Algorithms and Their Applications,* Pitts-burgh, PA, pp. 93C100. sponsored by Texas Instruments and U.S. Navy Center for Applied Research in Artificial Intelligence (NCARAI).

[FonsecaFleming1993] Fonseca, C. M. and P. J. Fleming (1993). Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In S. Forrest (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms,* San Mateo, California, pp. 416C423. Morgan Kaufmann.

[Horn1994] Horn, J., Nafpliotis, N. and Goldberg, D. E. (1994). A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Computation,* Volume 1, Piscataway, NJ, pp. 82C87. IEEE.

[SrinivasDeb1994] Srinivas, N. and Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary Computation 2(3), 221C248.

[Zitzler1999] Zitzler, E.(1999). Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. Ph. D. thesis, Swiss Federal Institute of Technology (ETH) Zurich, Switzerland. TIK-Schriftenreihe Nr. 30, Diss ETH No. 13398, Shaker Verlag, Aachen, Germany.

[Tiwari2002] Tiwari, A., Roy, R.(2002). Generalised Regression GA for Handling Inseparable Function Interaction: Algorithm and Applications. Proceedings of the seventh international conference on parallel problem solving from nature. (PPSN VII). Granada, Spain

[Deb2001a] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan(2001). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II, IEEE Transactions on Evolutionary Computation, Vol. 6, No. 2, pp. 182–197

[Zitzler2001] Zitzler, E., Laumanns, M. and Thiele, L. (2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm. TIK-Report 103. ETH Zentrum, Gloriastrasse 35, CH-8092 Zurich, Switxerland.

[PurshouseFleming2001] Purshouse, R. C., Fleming, P. J. (2001). The Multi-objective Genetic Algorithm Applied to Benchmark Problems–an Analysis. Research Report No. 796. Department of Automatic Control and Systems Engineering University of Sheffield, Sheffield, S1 3JD, UK.

[Knowles2000] Knowles, S. D. and Corne D. W.(2000). Approximating the nondominated front using the Pareto archived evolution strategy. Evolutionary Computation Journal, 8(2), 149-172.

[Kung1975] Kung, H. T., Luccio, F. and Preparata, F. P. (1975). On finding the maxima of a set of vectors. Jornal of the Association for Computing Machinery 22(4) 469-476.

[Deb2001b] Deb, K.(2001). Multi-objective optimization using evolutionary algorithms. Chichester, UK: Wiley