Performance Analysis of Path Relinking on Many-objective NK-Landscapes

Joseph M. Pasia,[†1,†2] Hernán Aguirre[†2]
and Kiyoshi Tanaka[†2]

Path relinking is a population-based heuristic that explores the trajectories in variable space between two elite solutions. It has been successfully used as a key component of several multi-objective optimizers, especially for solving bi-objective problems. In this paper, we focus on the behavior of pure path relinking, propose several variants of the path relinking that vary on their strategies of selecting solutions, and analyze its performance using several many-objective NK-landscapes as instances. The study shows that the path relinking becomes more effective in improving the convergence of the algorithm as we increase the number of objectives. It is also shown that the selection strategy associated to path relinking plays an important role to emphasize either convergence or spread of the algorithm.

1. Introduction

Multi-objective optimization (MO) is the process of simultaneously finding solutions to problems with two or more objectives. It is often called as *many-objective optimization* (MaO) if there are at least four objectives. MaO has attracted the interest of many researchers because of the poor performance of multi-objective evolutionary algorithms (MOEAs) that are known to be efficient in solving MO problems. Their poor performance is due to the large number of solutions in every Pareto front levels when the number of objectives is high[8], making their Pareto dominance ranking coarser, thus weakening their convergence property[1),5].

Most of the recent approaches that improve MOEAs introduce modifications that are focused mainly on the management of the objective space taking no or just slight consideration of the decision space. In this paper, we study the behavior of path relinking (PR), a procedure that provides a unifying principle for combining elite solutions to create new ones based on generalized path constructions in *both* objective and decision spaces[4]. Although the efficacy of path relinking in solving MO problems has been demonstrated, it has not been used as a stand-alone algorithm but only as key component of different optimizers. Moreover, except for 2) where it considered four-objective knapsack problems, it has not been applied to solve complex MaO problems. Thus, we propose several approaches for implementing path relinking for complex combinatorial optimization problems having many objectives. We also investigate how the different selection strategies associated to PR can emphasize either convergence or spread of the algorithm

To analyze the performance of PR on MO and MaO problems, we use the MNK-landscape models[1] as test instances. The MNK-landscape is an extension of Kauffman's NK-landscape models of epistatic interaction[7] to multi-objective *combinatorial* optimization problems. It is defined as a vector function mapping binary strings of length N into M real numbers $\boldsymbol{f} : \mathbf{Z}^N \to \mathbf{R}^M$, where $\mathbf{Z} = \{0, 1\}$. $\boldsymbol{K} = \{K_1, K_2, \ldots, K_M\}$ is a set of integers where each K_i gives the number of bits in the string that interact with each bit in the ith landscape. Each $f_i(\cdot)$ is expressed as

$$f_i(\boldsymbol{x}) = \frac{1}{N} \sum_{j=1}^{N} f_{i,j}(x_j, z_1^{(i,j)}, z_2^{(i,j)}, \ldots, z_{K_i}^{(i,j)}) \qquad (1)$$

where $f_{i,j} : \mathbf{Z}^{K_i+1} \to \mathbf{R}$ gives the fitness contribution of x_j to $f_i(\cdot)$, and $z_1^{(i,j)}, z_2^{(i,j)}, \ldots, z_{K_i}^{(i,j)}$ are the K_i bits interacting with x_j in string $\boldsymbol{x}$.

In this study, we consider landscapes with $2 \leq M \leq 10$ objectives, $N = 100$ bits, and $0 \leq K_i \leq 50$, $i = 1, 2 \ldots, M$ epistatic interactions. It is important to note that we do not aim to propose a pure PR as an alternative search procedure to MaO problems. Rather, we study the performance of PR on MNK-landscapes to provide useful insights for practitioners on how to exploit the desirable properties of PR to enhance existing MOEAs.

†1 Institute of Mathematics, University of the Philippines-Diliman, Quezon City, 1101 PHILIPPINES
†2 Faculty of Engineering, Shinshu University, 4-17-1 Wakasato, Nagano, 380-8553 JAPAN

2. Path Relinking for Many-Objective Optimization

Path relinking generates a sequence of solutions in the decision space by exploring the trajectories that connect elite solutions. Starting from an initiating solution ($\boldsymbol{i_s}$), it creates new solutions that form a *path* by performing moves in the decision space that progressively incorporate the attributes (e.g edges, nodes) of the guiding solution ($\boldsymbol{g_s}$)[4].

Like any other implementation of PR in multi-objective case, we perform PR between two solutions that belong to set $\mathcal{P}$ of potentially efficient solutions. However, since we deal primarily with many-objective problems, we consider several ways of defining the initiating and guiding solutions only from the set of solutions $\mathcal{P}' \subset \mathcal{P}$ that are considered best in each objective. Moreover, we use several forms of scalarizing functions to select new solutions to form the path. Whereas all applications perform local search procedures within PR to intensify the search towards the optimal Pareto front, we do not implement any such procedure in order to clearly reveal the behavior of the pure PR algorithm. **Figure 1** provides the algorithmic framework for the PR used in this study.

The procedure `Generate` creates a set of distinct random solutions and returns the nondominated solutions. The procedure `Define` initializes the set $(\mathcal{I},\mathcal{G})$ of $\boldsymbol{i_s}$–$\boldsymbol{g_s}$ pairs in every iteration by first determining the set $\mathcal{P}'$. Then, it sets the $\boldsymbol{i_s}$–$\boldsymbol{g_s}$ pairs via two proposed methods. The first method is called *Cycle*. Initially, *Cycle* arranges the solutions of $\mathcal{P}'$ in random order. Then, the first and second solutions are labeled as $\boldsymbol{i_s}$ and $\boldsymbol{g_s}$, respectively. For the succeeding pairs of solutions, the initiating solution is the guiding solution of the previous pair and the guiding solution is the next solution in $\mathcal{P}'$. The final $\boldsymbol{i_s}$-$\boldsymbol{g_s}$ pair are the last and first solutions in $\mathcal{P}'$, respectively. The second method called *Pair* simply forms a set of distinct pairs of different solutions from $\mathcal{P}'$ to serve as $\boldsymbol{i_s}$ and $\boldsymbol{g_s}$.

The actual generation of the sequence of solutions or *path* from solution $\boldsymbol{x}'$ to $\boldsymbol{g_s}$ consists of two procedures. The first procedure `PathRelink` returns at each step the set F of 1-bit neighbor solutions of $\boldsymbol{x}'$ that reduces the Hamming distance γ from $\boldsymbol{g_s}$, i.e. $F = \{\boldsymbol{x}'' \in \mathcal{N}(\boldsymbol{x}') : \gamma(\boldsymbol{x}'',\boldsymbol{g_s}) < \gamma(\boldsymbol{x}',\boldsymbol{g_s})\}$. `PathSelect` then immediately chooses a single solution from F having the best value of the

1: $\mathcal{P} \leftarrow$ `Generate()`;
2: **repeat** {/*iteration loop*/}
3:    $(\mathcal{I},\mathcal{G}) \leftarrow$ `Define`$(\mathcal{P})$;
4:    $\mathcal{S} \leftarrow \{\}$;
5:    **repeat**
6:      choose $(\boldsymbol{i_s},\boldsymbol{g_s}) \in (\mathcal{I},\mathcal{G})$;
7:      $\boldsymbol{x}' \leftarrow \boldsymbol{i_s}$;
8:      **repeat** {Path Generation}
9:        $F \leftarrow$ `PathRelink`$(\boldsymbol{x}',\boldsymbol{g_s},N,\gamma)$;
10:        $\boldsymbol{x}'' \leftarrow$ `PathSelect`(F,ω);
11:        $\mathcal{S} \leftarrow \mathcal{S} \cup \{\boldsymbol{x}''\}$;
12:        $\boldsymbol{x}' \leftarrow \boldsymbol{x}''$; $F = \{\ \}$;
13:      **until** $\gamma(\boldsymbol{x}',\boldsymbol{g_s}) < d_0$
14:      $(\mathcal{I},\mathcal{G}) \leftarrow (\mathcal{I},\mathcal{G})/\{(\boldsymbol{i_s},\boldsymbol{g_s})\}$;
15:    **until** $(\mathcal{I},\mathcal{G})$ is empty
16:    $\mathcal{P} \leftarrow$ Nondominated$(\mathcal{P} \cup \mathcal{S})$;
17:    $\mathcal{P} \leftarrow$ `Select`$(\mathcal{P},l)$;
18: **until** termination condition is satisfied
19: **return** nondominated set $\mathcal{P}$

Fig. 1 Path relinking algorithm

real-valued function ω expressed as the weighted sum fitness function by

$$\omega(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{f}(\boldsymbol{x}) \qquad (2)$$

where $\boldsymbol{w}=[w_1,w_2,\ldots,w_M]$ is a weight vector such that $\sum_{i=1}^{M} w_i = 1$ and $w_i \geq 0 \ \forall i$.

Initially, ω is defined as the objective function where the $\boldsymbol{g_s}$ is best. This strategy clearly prefers moves that are attractive relative to f_i. Moreover, it limits the search from the many objective standpoint to single objective optimization. The solution selected by `PathSelect` then becomes an intermediate solution of the path.

It is important to note that the two procedures have been expressed as a local search that optimizes a lexicographic objective function $\Phi = (\gamma,\omega)$[6]. Clearly, Φ constitutes two functions defined in the decision and objective spaces.

Table 1 Four variants of the path relinking algorithm. The symbol $i_s \leftrightarrow g_s$ ($i_s \rightarrow g_s$) indicates that there is (no) reversal of roles between initiating and guiding solutions

Cycle	$i_s \rightarrow g_s$	PRCycle1
	$i_s \leftrightarrow g_s$	PRCycle2
Pair	$i_s \rightarrow g_s$	PRPair1
	$i_s \leftrightarrow g_s$	PRPair2

Table 2 Selection strategies for the path relinking algorithm using PRCycle2 variant.

Fitness Function	Path relinking
$\omega(\boldsymbol{x}) = f_j(\boldsymbol{x})$	PRCycle2_w1.0
$\omega(\boldsymbol{x}) = 0.5f_i(\boldsymbol{x}) + 0.5f_j(\boldsymbol{x})$	PRCycle2_w0.5
$\omega(\boldsymbol{x}) = \sum_{i=1}^{M} w_i f_i(\boldsymbol{x})$	PRCycle2_aggr

In this study, we also consider re-initiating the process of path generation in the opposite direction by interchanging the roles of i_s and g_s, thus creating two paths. Combining the two strategies for setting the sets of i_s–g_s pairs, and whether to interchange the roles of i_s and g_s initially give us four variants of the PR algorithm. **Table 1** summarizes these variants.

Since the number of nondominated solutions in every Pareto front increases dramatically with M[1], the method `Select` performs the archiving by selecting the M extreme solutions of $\mathcal{P}$ and randomly selecting solutions from the remaining $l - M$ solutions. The value of l is set to 100.

3. Experimental Results and Analysis

We evaluate the performance of PR algorithms using the hypervolume $\mathcal{H}$ and coverage $\mathcal{C}$[9] metrics, the sum of maximum objectives $\mathcal{S}_{\max}$[5], and using the performance of conventional NSGA-II[3] as benchmark. The $\mathcal{H}$ metric uses several reference points $\mathcal{O}$ defined by the parameter α. If α equals zero then $\mathcal{O}$ is the origin $O = \{0, 0, \ldots, 0\}$, and as the value of α nears 1, $\mathcal{O}$ approaches the point W having the worst objective values as coordinates. If $\alpha = 0.5$, then $\mathcal{O}$ is the midpoint of the segment $\overline{OW}$. $\mathcal{S}_{\max}$ measures the convergence at the extremes and around the M edges of the Pareto front. It is given by $\mathcal{S}_{\max}(\mathcal{P}) = \sum_{i=1}^{M} \max_{\boldsymbol{x} \in \mathcal{P}} f_i(\boldsymbol{x})$.

3.1 Performance varying the number of objectives

We first analyze the performance of PR when M equals 2 to 10 and K_i is 7. The normalized $\mathcal{H}$ values or the ratio $\mathcal{H}(\text{PR})/\mathcal{H}(\text{NSGA-II})$, shows that NSGA-II outperforms the PR variants when $2 \leq M \leq 4$, and there is a decrease in $\mathcal{H}$ values when α increases to 0.99 and $2 \leq M \leq 3$. For example, **Fig. 2(a)** shows the normalize $\mathcal{H}$ metric for PRCycle2. Likewise, the $\mathcal{C}$ metric values in **Fig. 2(b)**

imply that NSGA-II weakly dominates all solutions of PR when $M = 2$ and it covers more solutions when $M = 3$. The extreme solutions of NSGA-II are also better compared to PR (see **Fig. 2(c)**). These results are expected since several studies have shown the effectiveness of NSGA-II in solving MO problems.

As M increases from 4 to 10, the convergence of PR variants improves. For example, when $M \geq 6$, at least 75% of the runs of PRCycle2 show improvement in $\mathcal{H}$ (i.e. normalized $\mathcal{H} > 1$) for all values of α. Also, although they only weakly dominate fewer solutions of NSGA-II, almost none of their solutions are covered by NSGA-II. On the other hand, the big difference in the performance in $\mathcal{H}$ between $\alpha = 0.5$ and $\alpha = 0.99$ can be attributed to the better convergence in the central regions.

Among the different PR variants, it can be seen that interchanging the roles of the i_s and g_s is beneficial only in terms of improving the extreme solutions for smaller number of objectives. It is shown that for $M \leq 5$, the median of the normalized $\mathcal{S}_{\max}$ values of PRCycle2 and PRPair2 are significantly better than that of PRCycle1 and PRPair1. Likewise, *Cycle* shows significant edge over *Pair* only in the $\mathcal{S}_{\max}$ metric and only between PRCycle1 and PRPair1.

3.2 Performance varying the selection

Since the weight vector $\boldsymbol{w}$ provides the direction for the search, we study three other ways of defining $\boldsymbol{w}$. First, we let $\boldsymbol{w}$ be a unit vector such that $w_j = 1$ if and only if i_s is best in f_j. Another method is to set w_i to 0.5 if and only if either the i_s or g_s is best in f_i, and zero otherwise. The final method targets the central region of the Pareto front by aggregating (aggr) all the objective functions. The values of the weights are changed for every call of `PathSelect`. **Table 2** shows the new PR variants based on the corresponding function ω.

Results show that the normalized $\mathcal{H}$ for the different selection strategies improves as M increases. Remarkably, PRCycle2_aggr posted the biggest improve-
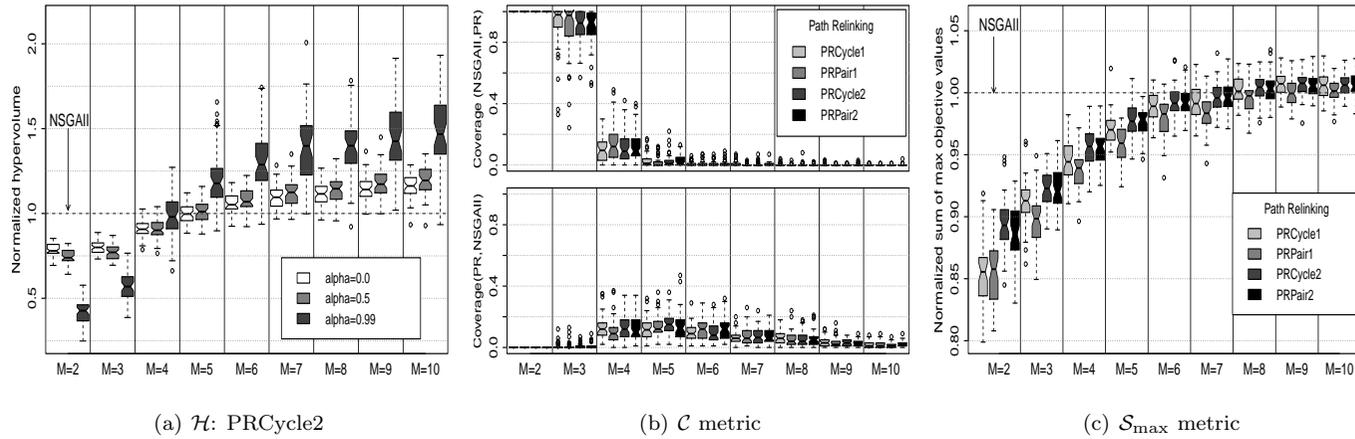
(a) $\mathcal{H}$: PRCycle2   (b) $\mathcal{C}$ metric   (c) $\mathcal{S}_{\max}$ metric

Fig. 2 (a) Normalized $\mathcal{H}$ metric for PRCycle2 (b) $\mathcal{C}$ metric and (c) normalized $\mathcal{S}_{\max}$ metric
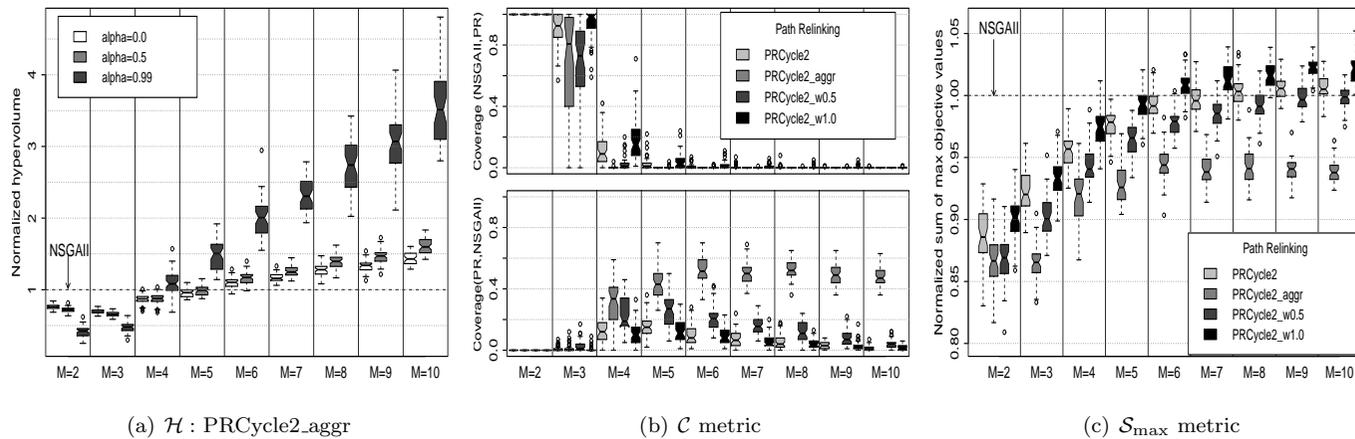between PR variants and NSGA-II for different M values and $K = 7$

(a) $\mathcal{H}$: PRCycle2_aggr   (b) $\mathcal{C}$ metric   (c) $\mathcal{S}_{\max}$ metric

Fig. 3 (a) Normalized $\mathcal{H}$ metric for PRCycle2_aggr (b) $\mathcal{C}$ metric and (c) normalized $\mathcal{S}_{\max}$ metric
between PR variants and NSGA-II for different M values and $K = 7$

ment (see **Fig. 3(a)**). The high $\mathcal{H}$ values translate to higher number of solutions of NSGA-II being weakly dominated by PRCycle2_aggr. Roughly between 40% to 60% of the solutions of NSGA-II are covered by PRCycle2_aggr when $M \geq 6$, while NSGA-II covers nothing of PRCycle2_aggr as shown in **Fig. 3(b)**.

The good convergence of PRCycle2_aggr as shown by its performance in $\mathcal{H}$ and $\mathcal{C}$ metrics expectedly sacrifices the quality of its extreme solutions and its solutions around the edges since the normalized $\mathcal{S}_{\max}$ metric in **Fig. 3(c)** shows that PRCycle2_aggr is totally outperformed by NSGA-II. It is PRCycle2 and PRCycle2_w1.0 that perform well in terms of $\mathcal{S}_{\max}$ with the latter obtaining the best extreme values. PRCycle_w1.0 even outperforms NSGA-II when $M \geq 6$. All these results suggest that the manner of selecting the intermediate solutions or creating the path is a valuable factor when implementing PR. The different selection strategies exhibit a trade-off between convergence and spread.

3.3 Performance varying the levels of epistastic interactions

To study the effects of varying K, we analyze the performances of PRCycle2 and PRCycle2_aggr when K ranges from 0 to 50 and under different M values. It can be observed in **Fig. 4(a)**–**Fig. 4(b)** that for all values of K, the $\mathcal{H}$ values of PR are better than NSGA-II only when M is high. However, the edge of PR over NSGA-II diminshes as K increases.

Figure 5(a) shows that NSGA-II covers almost all the solutions of PR for all K and $M = 2$. But, PRCycle2 and PRCycle2_aggr weakly dominated more solutions of NSGA-II than NSGA-II can cover them when $M > 4$. PRCycle2_aggr also has higher coverage of NSGA-II compared to PRCycle2. However, the coverage of PRCycle2 and PRCycle2_aggr over NSGA-II decreases as K increases. **Figure 5(b)** suggests that PRCycle2 obtains better extreme solutions than PRCycle2_aggr but both don't find extreme solutions that are good as NSGA-II. However, as K increases, there is an improving trend for the normalized $\mathcal{S}_{\max}$ of PRCycle2_aggr.

4. Conclusions

In this paper, we study the performance of a pure path relinking algorithm on MNK-landscape models having different number of objective functions and levels of epistatic interactions that define the complexity of the models. We
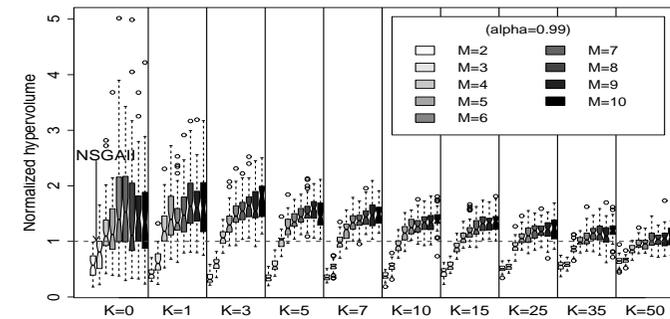
(a) $\mathcal{H}$: PRCycle2
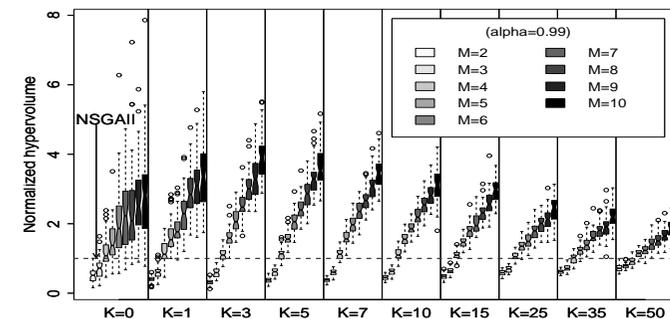
(b) $\mathcal{H}$: PRCycle2_aggr

Fig. 4 Normalized $\mathcal{H}$ metric for (a) PRCycle2 (b) PRCycle2_aggr under different values of M and K

design several variants of path relinking that differ on the way the initiating and guiding solutions are defined, and on whether to interchange their roles or not. In defining the initiating and guiding solutions, we propose two ways on how to obtain them from the set of potentially efficient solutions. Moreover, we study how the selection of the path of intermediate solutions using several fitness functions affects the performance of path relinking. Experiments show that the selection of pairs of initiating and guiding solutions and whether to interchange
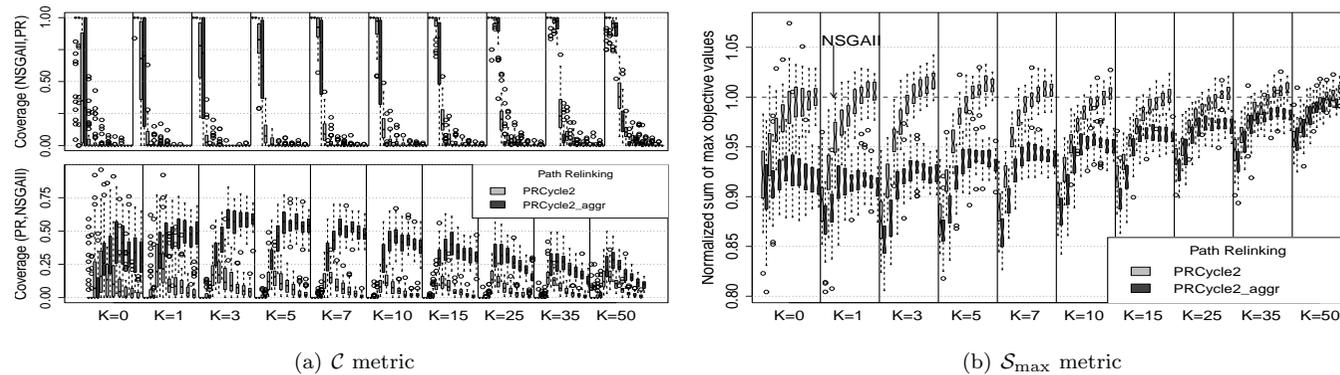
(a) $\mathcal{C}$ metric

(b) $\mathcal{S}_{\max}$ metric

Fig. 5 (a) $\mathcal{C}$ metric and (b) normalized $\mathcal{S}_{\max}$ metric between PR variants and NSGA-II
under different values of M and K

their roles or not have a minimal effect on the convergence around the extremes of the Pareto front. On the other hand, the selection of path can guide the search towards either the central region or the extremes of the Pareto front. However, path relinking exhibits a much stronger convergence property around the central region. In fact, it converges better than NSGA-II when $M \geq 4$. This good convergence can be seen in a broad range of level of epistatic interactions K, with its peak improvement around $1 \leq K \leq 10$.

In the future, we want to investigate adaptive strategies that simultaneously improve both convergence and spread. Likewise, we want to study the ways on how to enhance MOEAs using path relinking.

References

1) Aguirre, H.E. and Tanaka, K.: Working principles, behavior, and performance of MOEAs on MNK-landscapes, *European Journal of Operational Research*, Vol.181, No.3, pp.1670 – 1690 (2007).
2) Beausoleil, R.P., Baldoquin, G. and Montejo, R.: Multi-start and path relinking methods to deal with multiobjective knapsack problems, *Annals of Operations Research*, Vol.157, No.1, pp.105–133 (2008).
3) Deb, K., Agrawal, S., Pratap, A. and Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: NSGA-II, *PPSN*, pp.

849–858 (2000).
4) Glover, F., Laguna, M. and Marti, R.: Fundamentals of Scatter Search and Path Relinking, *Control and Cybernetics*, Vol.29, No.3, pp.653–684 (2000).
5) Ishibuchi, H., Tsukamoto, N. and Nojima, Y.: Evolutionary many-objective optimization: A short review, *Proc. of 2008 IEEE Congress on Evolutionary Computation*, Hong Kong, pp.2424–2431 (2008).
6) Jaszkiewicz, A. and Zielniewicz, P.: Pareto memetic algorithm with path relinking for bi-objective traveling salesperson problem, *European Journal of Operational Research*, Vol.193, No.3, pp.885 – 890 (2009).
7) Kauffman, S.A.: *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press (1993).
8) Khare, V., Yao, X. and Deb, K.: Performance scaling of multi-objective evolutionary algorithms, *Evolutionary Multi-Criterion Optimization*, Springer, pp.367–390 (2003).
9) Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. and Fonseca, V.: Performance assessment of multiobjective optimizers: An analysis and review, *IEEE Trans. Evolutionary Computation*, Vol.7, pp.117–132 (2003).