Pareto Rank Learning in Multi-objective Evolutionary Algorithms

Chun-Wei Seah, Yew-Soon Ong, Ivor W. Tsang, Siwei Jiang
School of Computer Engineering
Nanyang Technological University
Singapore 639798
Email:{seah0116,asysong,ivortsang,sjiang1}@ntu.edu.sg

***Abstract*—In this paper, the interest is on cases where assessing the goodness of a solution for the problem is costly or hazardous to construct or extremely computationally intensive to compute. We label such category of problems as "expensive" in the present study. In the context of multi-objective evolutionary optimizations, the challenge amplifies, since multiple criteria assessments, each defined by an "expensive" objective is necessary and it is desirable to obtain the Pareto-optimal solution set under a limited resource budget. To address this issue, we propose a Pareto Rank Learning scheme that predicts the Pareto front rank of the offspring in MOEAs, in place of the "expensive" objectives when assessing the population of solutions. Experimental study on 19 standard multi-objective benchmark test problems concludes that Pareto rank learning enhanced MOEA led to significant speedup over the state-of-the-art NSGA-II, MOEA/D and SPEA2.**

***Keywords*-Multi-objective Evolutionary Algorithms, Expensive Problems, Pareto Rank Learning**

I. Introduction

Modern stochastic optimization, such as Multi-objective Evolutionary Algorithms (MOEAs) in particular, has gained increasing popularity in the past decades. From a survey of the field [1], there has been notable increasing trends on the research and development of novel evolutionary frameworks and algorithms, suitable for handling complex problems more effectively and efficiently. However, general MOEAs often require significant number of fitness evaluations before the reproduction mechanisms take effect to arrive at near Pareto-optimal solutions. Recent study on MOEAs [2], specifically NSGA-II [3] and SPEA2 [4], have shown that it takes approximately 10 to 140 thousands fitness evaluations for each objective on the ZDT 32-variables test problems, to converge at Pareto solutions with 98% hypervolume of the true Pareto front.

In many of today's realistic applications in science and engineering, it is often the case that the fitness value of a potential solution may be difficult to obtain. It may happen that the assessment or evaluation of the solution is not always readily available, too costly or hazardous to be constructed or the assessment is extremely computationally intensive. We label such problems as "expensive" in the present paper. A motivating example is in massively multiplayer online game, where large number of expensive Monte-Carlo simulations is often necessary to assess or evaluate the potential game states and solutions involving multiple criteria by actually playing out the game to its end via a series of moves. Hence, every evaluation is often very computationally intensive [5]. Other "expensive" applications include those encountered in aerodynamic and structural design [6], energy minimization of the biomolecule [7], economics and finance (where costly human expert involvements to evaluate the potential solutions are required) [8], etc.

When dealing with "expensive" problems, minimizing the number of assessments or evaluations is often desirable. Considering an "expensive" single-objective or single-criterion optimization problem takes a minute to assess a potential solution at some figure of merit, often known as the fitness value. For 10 thousands evaluations, an approximation of 7 days would be necessary to complete the search, while 140 thousands of fitness evaluations would take 3 months to complete. The complexity amplifies for multi-criteria assessments, with the computational effort required intensifying to approximately $7m$ days - $3m$ months on m-objectives optimization problem. This results in a major impediment on the use of evolutionary algorithm for time conscious problems and "expensive" problems. In summary, on "expensive" problems, classical MOEAs do not cope elegantly.

To date, a popular way to enhance the search efficiency of evolutionary algorithm when dealing with "expensive" problems is to use computationally cheap approximation or surrogate models in place of the original assessment option [9], [10]. In the last decade, research studies on data-centric surrogate-assisted optimizations have thrived, mainly as a result of the significant increase in the number of "expensive" real world complex applications that have emerged. In multi-objective evolutionary optimizations, the importance and challenges of using surrogates amplify with increasing objectives [2], since the multiple criteria assessments, each defined by an "expensive" objective function is necessary, while it is desirable to obtain the Pareto-optimal solution set under limited computational budget.

Despite the increased research efforts, the study on existing surrogate-assisted evolutionary frameworks remain open for further improvements. Particularly, present surrogate-assisted evolutionary frameworks proposed are typically flawed by the introduction of false optima, since it is difficult to model the "expensive" problem accurately due to the challenges brought

about by the *curse of dimensionality*, thus leading to possibly unreliable search. With this in mind, recent studies [6], [11] are placing greater emphasis on the importance of achieving improvements in solution quality, as opposed to the usual practice of improving the regression accuracy of the surrogate model in the context of evolutionary optimization since the latter remains a hard problem. Some have also turned to favor using binary classification models over regression models as the surrogate used [7] when dealing with "expensive" non-linear constrained programming problem.

Our core interest in the present study is on data-centric surrogate-assisted multi-objective evolutionary search methodologies in the spirit of *Optinformatics*[1]. From a survey of the MOEA literature, it is worth noting that many of the state-of-the-art multi-objective evolutionary algorithms including the NSGA-II, SPEA2, SMPSO [1], etc., use the relative Pareto front rank information from the solution fitness values obtained to effect their search. Moreover, unlike regression models, which easily deteriorate when inaccurate numeric outputs are given, fitting the ordinal scale outputs is more robust and helps identify the Pareto rank information of solutions. Taking this cue, this paper focuses on surrogates that directly predict the Pareto front ranks of the solutions in the offspring, as opposed to regress the fitness values [13], [14], [15], [16], since the former problem is easier to address than estimating the true fitness value precisely. In contrast to previous studies that build a regression model for each separate objective, and subsequently using the independently predicted objective values (or aggregating the objectives into single value via parameterized scalarizing weight vector) within the surrogate-assisted multi-objective evolutionary search [15], [6], we propose here a *Pareto Rank Learning* (PRL) scheme to learn and predict the Pareto front ranks of the offspring for the multiple objectives simultaneously. Using the predicted Pareto front rank of the offspring, selection of the new population can then be achieved at significantly lower number of exact evaluation calls to the multiple "expensive" objectives during the evolution, thus leading to significant cost savings to arrive at near true Pareto front.

The rest of this paper is organized as follows: Section II presents some preliminaries on multi-objective evolutionary optimization and introduces the proposed Pareto Rank Learning scheme for enhancing multi-objective evolutionary search on "expensive" problems. Extensive experiments on 19 test problems are carried out by comparing several state-of-the-art algorithms with the PRL enhanced MOEA on three performance metrics and the experimental results are analyzed and discussed in Section III. Lastly, the brief conclusion of this paper is drawn in Section IV.

[1]*Optinformatics* is the specialization of informatics for the processing of the data generated in optimization in order to extract possibly implicit and potentially useful information and knowledge, which could be helpful for understanding search mechanism of the solver, guiding/improving search, and/or revealing undisclosed insights to the problem structure [12].

II. Proposed Study

In this section, we present a brief preliminary of the multi-objective evolutionary optimization problem and the detailed steps of the proposed Pareto Rank Learning scheme.

A. Preliminaries on Multi-objective Evolutionary Optimization

Without loss of generality, we consider here the multi-objective evolutionary optimization of $\mathbf{f}$ with m objectives $(f_1, \ldots, f_m)$ that takes the form:

$$\begin{aligned} \textbf{Minimize}: \quad & \mathbf{f}(\mathbf{s}) = (f_1(\mathbf{s}), \ldots, f_m(\mathbf{s})) \\ \textbf{Subject to}: \quad & \mathbf{s}_l \leq \mathbf{s} \leq \mathbf{s}_u \end{aligned} \tag{1}$$

where $\mathbf{s} \in R^d$ is the vector of optimization variables, and $\mathbf{s}_l$, and $\mathbf{s}_u$ are vectors of lower and upper bounds, respectively. If $\mathbf{f}(\mathbf{s_1}) = (f_1(\mathbf{s}_1), \cdots, f_m(\mathbf{s}_1))$ and $\mathbf{f}(\mathbf{s_2}) = (f_1(\mathbf{s}_2), \cdots, f_m(\mathbf{s}_2))$ denote two solution vectors, $\mathbf{s}_1$ is said to *dominate* $\mathbf{s}_2$, as notated by $\mathbf{s}_1 \succ \mathbf{s}_2$ iff $\forall i : f_i(\mathbf{s}_1) \leq f_i(\mathbf{s}_2)$ and $\exists i : f_i(\mathbf{s}_1) < f_i(\mathbf{s}_2)$. The set of N solutions $S = \{\mathbf{s}_1, \cdots, \mathbf{s}_N\}$ can then be sorted according to their Pareto front rank $\mathcal{F} = \{\mathcal{F}_1, \cdots, \mathcal{F}_k, \cdots, \mathcal{F}_K\}$ with the following satisfying conditions:

$$\begin{aligned} &1) \forall \mathbf{s}_i, \mathbf{s}_j \in \mathcal{F}_k : \mathbf{s}_i \not\succ \mathbf{s}_j \wedge \mathbf{s}_j \not\succ \mathbf{s}_i \\ &2) \forall \mathbf{s}_i \in \mathcal{F}_{k+1}, \forall \mathbf{s}_j \in \mathcal{F}_k : \mathbf{s}_i \not\succ \mathbf{s}_j \\ &3) \forall \mathbf{s}_i \in \mathcal{F}_{k+1}, \exists \mathbf{s}_j \in \mathcal{F}_k : \mathbf{s}_j \succ \mathbf{s}_i \end{aligned} \tag{2}$$

where $\sum_{i=1}^{K} |\mathcal{F}_i| = N$ and K is the resultant number of Pareto front rank after sorting S. If r_i denotes the Pareto front rank of $\mathbf{s}_i$, then $\forall \mathbf{s}_j \in \mathcal{F}_k \implies r_j = k$.

The basic workflow of a classical MOEA is depicted in Figure 1(a), while the detailed pseudo-code is given in Algorithm 1. The classical MOEA search begins with the initialization of a population of potential solutions or individuals (see process 1 of Figure 1(a) and line 2 of Algorithm 1), either randomly or using design of experiments methods such as via Latin hypercube sampling. The population then proceeds with the MOEA operators of crossover and mutation (see process 2 of Figure 1(a) and line 5 of Algorithm 1). All solution individuals generated, known as offspring, then undergoes fitness evaluations using the original objective functions $\mathbf{f}$ (see process 3a of Figure 1(a) and line 17 of Algorithm 1). In addition, all unique individual solutions and their associated exact fitness values of $\mathbf{f}$ obtained during the MOEA search are archived in a centralized database $\mathcal{D}$. Using the fitness values associated with each individuals in the population, the subsequent population is formed via the MOEA selection operator (see process 4 of Figure 1(a) and line 21 of Algorithm 1). In Pareto dominance MOEA, the Pareto front rank of each individual is used as one of the criteria in the selection operator, as defined by Eqn. (2). This process of MOEA search then continues until the computational budget set is exhausted or some other user specified termination criteria are met. When the algorithm ends, the non-dominated solutions are attained (see process 5 in Figure 1(a) and line 24 of Algorithm 1).

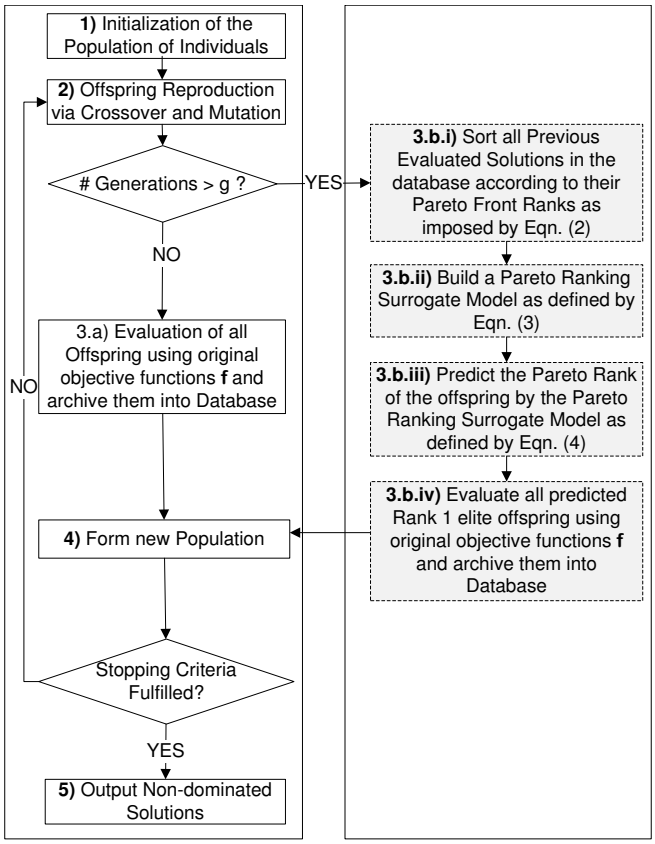

Fig. 1: Workflow of Pareto Rank Learning Multi-objective Evolutionary Algorithms. Subfigures (a) and (b) depicts the processes of the classical MOEA and Pareto Rank Learning scheme, respectively.

Algorithm 1 Pareto Rank Learning (PRL) scheme in a classical MOEA

Set t=0 (Generation counter)
An initial population $\mathcal{P}_t$ is generated and then evaluated using $\mathbf{f}(.)$.
$\mathcal{D} = \mathcal{P}_t$ (an archive to store all evaluated solutions)
while Stopping criteria is not reached **do**
  Perform crossover and mutation to create new population $\mathcal{Q}_t$ from $\mathcal{P}_t$
  $\mathcal{E}_t = \emptyset$ (evaluated $(\mathbf{s}, \mathbf{f}(\mathbf{s}))$ solution in generation t)
  if $t >= g$ (i.e., size of the archive has amply evaluated solution samples for reliable training) **then**
    sort $\mathcal{D}$ into independent Pareto front ranks $\mathcal{F} = \{\mathcal{F}_1, \cdots, \mathcal{F}_k, \cdots, \mathcal{F}_K\}$, see Eqn. (2)
    Construct Pareto ranking model $\mathcal{M}(\mathcal{F})$
    for all $\mathcal{Q}_t$ **do**
      if $\mathcal{Q}_t^i$ is predicted as rank one by $\mathcal{M}$, i.e., $\hat{r} = 1$ in Eqn. (4) **then**
        evaluate $\mathcal{Q}_t^i$ using $\mathbf{f}(.)$
        $\mathcal{E}_t = \mathcal{E}_t \bigcup \mathcal{Q}_t^i$
      end if
    end for
  else
    evaluate $\mathcal{Q}_t$ using $\mathbf{f}(.)$
    $\mathcal{E}_t = \mathcal{Q}_t$
  end if
  $\mathcal{D} = \mathcal{D} \bigcup \mathcal{E}_t$
  $\mathcal{P}_{t+1}$=select the next population from $\mathcal{E}_t \bigcup \mathcal{P}_t$
  t=t+1
end while
return Non-dominated solutions of P_t

B. Pareto Rank Learning in Multi-objective Evolutionary Algorithms

In this paper, we are interested in cases where the assessment of $\mathbf{f}(\mathbf{s})$ is "expensive", and it is desirable to obtain a near true-Pareto optimal solution under limited computational budget. The workflow of the MOEAs with the proposed Pareto Rank Learning scheme embedded, to enhance the efficiency of the evolutionary search on "expensive" problems, is depicted by process 3b(i, ii, iii and iv) of Figure 1(b).

For the sake of readability, we outline the Pareto Rank Learning Multi-objective Evolutionary Algorithms as three subsections, with the first focusing on the general workflow of the Pareto rank learning scheme, before proceeding to the details of the surrogate modelling algorithm and implementations and lastly an example of the Pareto ranking surrogate in MOEA at work is illustrated.

1) Workflow of Pareto Rank Learning: After some generations of performing the classical MOEA search, specifically when ample data points for surrogate building have been collated in database $\mathcal{D}$, Pareto Rank Learning kicks in, where learning and construction of surrogate models that approximate the Pareto front ranks for the solutions of the "expensive" problem begin. These surrogates are statistical models that are many orders of magnitude cheaper to run and used in lieu of the "expensive" problem during the MOEA search. In contrast to previous works on surrogate-assisted MOEAs for solving "expensive" problems as stated earlier, the core novelty of our work here lies in the construction of surrogates that directly model the Pareto front ranks of the offspring, in favor of those efforts that tried to regress the fitness values [13], [14], [15], [16].

Subsequently, the constructed Pareto Ranking surrogate model is used to assess and categorize individuals of the offspring into their respective Pareto front rank. The predictions produced by the surrogate model thus serves to pre-screen the offspring such that only the elite solution individuals, i.e., those offspring that are grouped at higher rank(s) are more worthy to undergo exact assessments or evaluations based on the original "expensive" objectives $\mathbf{f}$. This reduces the unnecessary "expensive" evaluations of inferior offspring whose Pareto front ranks in the population are anticipated to be poor, thus arriving at near the true Pareto front more efficiently. In the present study, we employ an elitism scheme where only solutions inferred as Pareto front rank 1 would undergo the

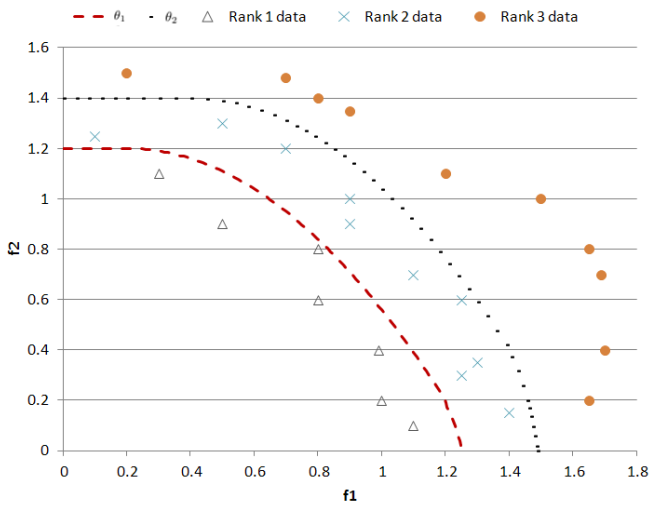

Fig. 2: A sorted archived solutions of a two-objectives problem denoted by f1 and f2 on the x-axis and y-axis, respectively, with 3 Pareto front ranks, i.e., $K = 3$ for $\mathcal{F} = (\mathcal{F}_1, ..., \mathcal{F}_K)$. The dotted curves denote the decision boundaries defining the Pareto front ranks as modelled in the Pareto ranking surrogate.

"expensive" evaluations. These evaluated elite offspring are subsequently combined with the population of parent solution individuals to generate the new population of individuals via the selection operation. In the next subsection, the algorithm for building Pareto ranking surrogate models is described.

2) Pareto Ranking Surrogate Modelling Algorithm and Implementations: For the purpose of brevity, the algorithm of the proposed Pareto Rank Learning scheme is highlighted using underlines in Algorithm 1. In the initial step of the proposed Pareto Rank Learning scheme, all archived solutions in database $\mathcal{D}$ are first sorted into independent Pareto front ranks $\mathcal{F} = \{\mathcal{F}_1, \cdots, \mathcal{F}_k, \cdots, \mathcal{F}_K\}$ accordingly to Eqn. (2). In this paper, we considered the non-dominance sorting method in [3] to obtain the Pareto front ranks of the archived solutions $\mathcal{D}$.

Using the sorted archived solutions $\mathcal{F}$, a Pareto Rank surrogate model $\mathcal{M}$ is subsequently built. The backbone of our scheme includes a statistical learning algorithm to build the surrogate $\mathcal{M}$ from $\mathcal{F}$ that models the Pareto front ranks of the offspring for each MOEA search generation. If $\mathbf{s}_i$ denotes the ith solution of the sorted solutions set $\mathcal{F} = \{\mathcal{F}_1, \cdots, \mathcal{F}_k, \cdots, \mathcal{F}_K\}$ according to their k Pareto front rank (i.e., as denoted in line 8 of Algorithm 1) and r_i denotes the Pareto front rank of $\mathbf{s}_i$, without loss of generality, we describe here the process for constructing the Pareto ranking surrogate model based on the well-established Support Vector Machine (SVM) [17], [18], [19], [20]. In particular, we considered Ordinal SVM surrogate models here [17] (although it is worth noting that other forms of learning machines may also apply), which takes the following form:

$$\begin{array}{rl} \min_{\mathbf{w},\boldsymbol{\theta}} & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{2}\|\boldsymbol{\theta}\|^2 + C\sum_{i=1}^{N} \ell(h(\mathbf{s}_i), r_i, \boldsymbol{\theta}) \\ s.t & \theta_0 < \theta_1 < ... < \theta_k - 1 < \theta_K \end{array} \quad (3)$$

where $\mathbf{w} \in \Re^d$ are the surrogate model variables, θ_k is the decision boundaries between rank k and $k+1$ of the K Pareto front. θ_0 and θ_K are defined as $-\infty$ and $+\infty$, respectively, C is the parameter that trade-offs the amount of regularization against the loss function $\ell(\cdot)$ and $h(\mathbf{s}_i) = \mathbf{w}'\mathbf{s}_i$. Using the Pareto front surrogate model, the rank $\hat{r}$ of each given offspring $\mathbf{s}$ is predicted as:

$$\hat{r} = \sum_{k=0}^{K} I[h(\mathbf{s}) \geq \theta_k] \quad (4)$$

where $I[\cdot]$ is an indicator function which returns one if the predicate holds, otherwise a zero is returned. Thus an output of rank k is given when $h(\mathbf{s})$ falls between θ_{k-1} and θ_k.

3) An Illustration Example of the Pareto Ranking Surrogate in MOEA at Work: To illustrate the course of actions by the Pareto ranking surrogate in a MOEA search, we consider the example of a two-objectives evolutionary problem in Figure 2 where the sorted archived solutions of database D at t generation are categorized into three Pareto front ranks. A Pareto ranking surrogate is then built, with the offspring solutions lying below the θ_1 decision boundary modeled to produce an output of rank 1, solutions between θ_1 and θ_2 decision boundaries as rank 2, while those above θ_2 decision boundary modeled as rank 3. Using the trained Pareto ranking surrogate model, the Pareto front rank of the offspring at $t+1$ generation of the MOEA search can then be inferred without calls to the "expensive" objectives $\mathbf{f}(.)$.

III. EXPERIMENTAL STUDY

In the present section, the proposed **Pareto Rank Learning** scheme is used to enhance the classical NSGA-II [3] and investigated on 19 benchmark multi-objective problems[2] (MOPs). Numerical studies of the scheme is then performed by pitting the Pareto rank learning NSGA-II against three state-of-the-art baseline algorithms, namely SPEA2 [4], MOEA/D [21] and the classical NSGA-II [3]. The set of benchmark problems includes UF1 to UF7, ZDT1 to ZDT6 except ZDT5 and DTLZ1 to DTLZ7 [22]. For the sake of consistency, the experimental studies of all algorithms are conducted on a common platform based on jMetal 3.1 [22].

In the present study, the population size is configured to 100. Independent studies for limited computational budgets of 2000 and 4000 evaluations are considered. In order to obtain statistically significant results, 30 independent runs are conducted on each test problem. To facilitate a fair comparison between the algorithms, common seeds in the random number generator are used for the algorithms in comparison across all the 30 independent runs. Hence, the classical NSGA-II and Pareto rank learning enhanced NSGA-II are ensured to share a common set of non-dominated solutions in the first g generations of each run, before the effects of the Pareto

[2]http://jmetal.sourceforge.net

rank learning scheme begins to bite. In the experimental study, g is initialized to 10, and the Pareto Ranking model is built using the ordinal SVM algorithm described in [17] with Gaussian kernel and other default parameters[3]. Three performance metrics, i.e., Generational Distance, Inverted Generational Distance and Hypervolume, are reported in the experimental results. The reported results are the mean values of each performance metric and the statistical significance are then carried out using statistical pairwise t-test.

A. Results of PRL Enhanced MOEA and Baseline Algorithms for Computational Budget of 2000 Evaluations

The detailed search performances of the Pareto Rank Learning enhanced NSGA-II and state-of-the-art baseline algorithms in terms of Generational Distance, Inverted Generational Distance and Hypervolume are tabulated in Tables I, II and III, respectively. Note that lower values of Generational Distance and Inverted Generational Distance are desirable while a higher value for Hypervolume is desired.

From the results in Tables I, II and III, it can be observed that all three baseline algorithms, i.e., SPEA2, MOEA/D and NSGA-II, fared significantly poorer than the PRL enhanced NSGA-II on at least 12 out of the 19 problems considered. For the sake of conciseness and precision, Table IV summaries the number of problems that a particular algorithm is significantly better or poorer, or insignificantly better or poorer (competitive) relative to another algorithm in terms of Generational Distance (see sub-Table (a)), Inverted Generational Distance (see sub-Table (b)) and Hypervolume (see sub-Table (c)). Overall, the PRL enhanced NSGA-II is shown to significantly outperform SPEA2 on 19/19, 18/19, 13/19, MOEA/D on 16/19, 15/19, 13/19, and the classical NSGA-II on 18/19, 15/19, 12/19, of the problems in terms of Generational Distance, Inverted Generational Distance and Hypervolume, respectively, under a limited computational budget of 2000 evaluations.

Furthermore, under a limited computational budget of evaluations, the details results in I, II and III and the summarized results in Table IV indicated that in overall the classical NSGA-II performed significantly better than SPEA2 on 15, 15, 9 and MOEA/D on 15, 12, 10 out of the 19 problems considered for Generational Distance, Inverted Generational Distance and Hypervolume, respectively. Taking this cue, in the next subsection, we perform a detailed analysis of the proposed Pareto Rank Learning enhanced NSGA-II and the classical NSGA-II for a limited computational budget of up to 4000 exact evaluations.

B. Analysis of PRL and NSGA-II Enhanced NSGA-II with a Computational Budget of up to 4000 evaluations.

Here, we present a detailed analysis on the convergence of NSGA-II and PRL enhanced NSGA-II for a computational budget of up to 4000 exact evaluations and showcases the different stages of the MOEAs search. Recall that the Pareto Rank Learning scheme begins to take effect only after 10 generations (1000 exact evaluations), since $g = 10$ as stated in the experimental setup. Thus the classical NSGA-II and PRL enhanced NSGA-II share common initial populations in the first 1000 evaluations since common seeds in the random number generator are employed. Thus we summarize in Figure 3 the percentage of problems that the PRL enhanced NSGA-II outperforms significantly or insignificantly better or under-performs significantly over the classical NSGA-II, for

[3]http://www.work.caltech.edu/∼htlin/program/libsvm/#ordinal

TABLE I: Results of *Generational Distance* at 2000 evaluations. $+$, $\approx$ and $-$ denote whether the algorithm is significantly better, insignificantly better or poorer (competitive) and significantly poorer, respectively, than the PRL enhanced NSGA-II. Each cell is represented as V(S)X where V, S and X are the mean value of the Generational Distance, standard deviation value and significant symbols (i.e., $+$, $\approx$ or $-$), respectively. PRL denotes Pareto Rank Learning and it enhanced on NSGA-II. The bold values indicate PRL enhanced NSGA-II is significantly better than all three baseline algorithms considered.

	SPEA2	MOEA/D	NSGA-II	PRL
UF1	0.0158(0.0021)−	0.0290(0.0035)−	0.0151(0.0021)−	**0.0044(0.0011)**
UF2	0.0257(0.0065)−	0.0432(0.0046)−	0.0229(0.0049)−	**0.0120(0.0030)**
UF3	0.0739(0.0100)−	0.0647(0.0152)−	0.0750(0.0134)−	**0.0533(0.0054)**
UF4	0.0204(0.0003)−	0.0209(0.0007)−	0.0198(0.0003)−	**0.0178(0.0008)**
UF5	0.2234(0.0226)−	0.2960(0.0158)−	0.2102(0.0253)−	**0.1443(0.0337)**
UF6	0.0962(0.0141)−	0.1491(0.0154)−	0.0917(0.0104)−	**0.0558(0.0062)**
UF7	0.0165(0.0019)−	0.0308(0.0041)−	0.0143(0.0025)−	**0.0053(0.0010)**
ZDT1	0.1269(0.0204)−	0.1728(0.0169)−	0.0844(0.0095)−	**0.0492(0.0173)**
ZDT2	0.1874(0.0207)−	0.2728(0.0217)−	0.1410(0.0160)−	**0.0695(0.0148)**
ZDT3	0.0711(0.0110)−	0.1015(0.0088)−	0.0507(0.0049)−	**0.0255(0.0074)**
ZDT4	5.6932(1.0150)−	6.6413(0.2142)−	4.4889(0.6677)−	**3.8732(0.8232)**
ZDT6	0.5277(0.0312)−	0.6067(0.0635)−	0.4613(0.0346)−	**0.3784(0.0380)**
DTLZ1	16.8067(3.893)−	11.8117(0.851)≈	13.5875(2.764)≈	13.0065(3.157)
DTLZ2	0.0072(0.0024)−	0.0060(0.0015)−	0.0043(0.0010)−	**0.0029(0.0026)**
DTLZ3	42.5490(6.006)−	28.5173(2.012)≈	33.6214(3.693)−	30.2788(4.387)
DTLZ4	0.0067(0.0028)−	0.0093(0.0029)−	0.0052(0.0023)−	**0.0017(0.0010)**
DTLZ5	0.0073(0.0024)−	0.0060(0.0015)−	0.0043(0.0010)−	**0.0028(0.0026)**
DTLZ6	0.7794(0.0178)−	0.4033(0.0491)+	0.7217(0.0183)−	0.6255(0.0190)
DTLZ7	0.1571(0.0238)−	0.2440(0.0272)−	0.1078(0.0132)−	**0.0541(0.0181)**

TABLE II: Results of *Inverted Generational Distance* at 2000 evaluations. Refer to Table I for more detailed description.

	SPEA2	MOEA/D	NSGA-II	PRL
UF1	0.0049(0.0003)−	0.0085(0.0010)−	0.0046(0.0003)−	**0.0043(0.0003)**
UF2	0.0053(0.0004)−	0.0106(0.0011)−	0.0048(0.0004)−	**0.0035(0.0002)**
UF3	0.0194(0.0009)−	0.0186(0.0006)−	0.0195(0.0015)−	**0.0165(0.0007)**
UF4	0.0058(0.0002)−	0.0063(0.0002)−	0.0056(0.0001)−	**0.0049(0.0003)**
UF5	0.3797(0.0415)−	0.5124(0.0513)−	0.3449(0.0517)−	**0.2693(0.0510)**
UF6	0.0243(0.0024)−	0.0411(0.0039)−	0.0227(0.0020)−	**0.0178(0.0013)**
UF7	0.0050(0.0004)−	0.0088(0.0012)−	0.0043(0.0004)−	**0.0034(0.0003)**
ZDT1	0.0245(0.0028)−	0.0536(0.0077)−	0.0182(0.0021)−	**0.0092(0.0019)**
ZDT2	0.0505(0.0059)−	0.0913(0.0066)−	0.0393(0.0058)−	**0.0275(0.0096)**
ZDT3	0.0280(0.0028)−	0.0589(0.0062)−	0.0212(0.0019)−	**0.0112(0.0016)**
ZDT4	0.6374(0.1563)−	1.0307(0.2089)−	0.4619(0.1501)≈	0.4073(0.0808)
ZDT6	0.1605(0.0127)−	0.1904(0.0288)−	0.1465(0.0113)−	**0.1185(0.0142)**
DTLZ1	2.2078(0.6521)≈	1.1496(0.9961)+	2.1279(0.5112)≈	2.2399(0.5759)
DTLZ2	0.0046(0.0012)−	0.0031(0.0007)−	0.0029(0.0007)−	**0.0019(0.0008)**
DTLZ3	10.2736(2.253)−	5.5749(3.5376)+	9.5110(1.6894)≈	9.0451(1.4963)
DTLZ4	0.0037(0.0028)−	0.0027(0.0009)≈	0.0025(0.0026)≈	0.0021(0.0028)
DTLZ5	0.0046(0.0012)−	0.0031(0.0007)−	0.0029(0.0007)−	**0.0019(0.0008)**
DTLZ6	0.5073(0.0172)−	0.2095(0.0442)+	0.4703(0.0166)−	0.4006(0.0144)
DTLZ7	0.0440(0.0061)−	0.1090(0.0159)−	0.0361(0.0037)−	**0.0176(0.0052)**

TABLE III: Results of *Hypervolume* at 2000 evaluations. Refer to Table I for more detailed description.

	SPEA2	MOEA/D	NSGA-II	PRL
UF1	0.4053(0.0138)—	0.2841(0.0294)—	0.4214(0.0143)—	**0.4573(0.0116)**
UF2	0.4121(0.0208)—	0.2305(0.0321)—	0.4391(0.0174)—	**0.5232(0.0145)**
UF3	0.0487(0.0174)—	0.0595(0.0105)—	0.0495(0.0239)—	**0.1031(0.0191)**
UF4	0.1145(0.0024)—	0.1009(0.0037)—	0.1170(0.0029)—	**0.1318(0.0056)**
UF5	0.0000(0.0000)≈	0.0000(0.0000)≈	0.0000(0.0000)≈	0.0000(0.0000)
UF6	0.0000(0.0000)—	0.0000(0.0000)—	0.0001(0.0005)—	**0.0035(0.0070)**
UF7	0.2700(0.0166)—	0.1284(0.0347)—	0.2953(0.0188)—	**0.3537(0.0126)**
ZDT1	0.0107(0.0138)—	0.0000(0.0000)—	0.0645(0.0312)—	**0.2854(0.0554)**
ZDT2	0.0000(0.0000)—	0.0000(0.0000)—	0.0000(0.0000)—	**0.0050(0.0098)**
ZDT3	0.0560(0.0236)—	0.0000(0.0002)—	0.1254(0.0245)—	**0.2847(0.0236)**
ZDT4	0.0000(0.0000)≈	0.0000(0.0000)≈	0.0000(0.0000)≈	0.0000(0.0000)
ZDT6	0.0000(0.0000)≈	0.0000(0.0000)≈	0.0000(0.0000)≈	0.0000(0.0000)
DTLZ1	0.0000(0.0000)≈	0.0350(0.0836)+	0.0000(0.0000)≈	0.0000(0.0000)
DTLZ2	0.1515(0.0135)—	0.1596(0.0108)—	0.1670(0.0082)—	**0.1929(0.0051)**
DTLZ3	0.0000(0.0000)≈	0.0011(0.0061)≈	0.0000(0.0000)≈	0.0000(0.0000)
DTLZ4	0.0749(0.0677)—	0.0279(0.0315)—	0.1064(0.0662)≈	0.1386(0.0851)
DTLZ5	0.1514(0.0136)—	0.1596(0.0108)—	0.1669(0.0081)—	**0.1927(0.0050)**
DTLZ6	0.0000(0.0000)≈	0.0000(0.0000)≈	0.0000(0.0000)≈	0.0000(0.0000)
DTLZ7	0.0000(0.0000)—	0.0000(0.0000)—	0.0012(0.0022)—	**0.0533(0.0234)**

TABLE IV: Number of test problems that a particular algorithm is significantly better or poorer using various performance metrics at 2000 evaluations. W, D and L denote the number of test problems that the algorithm on the column is significantly better, insignificantly better or poorer (competitive) and significantly poorer, respectively, than the algorithm on the row. PRL denotes Pareto Rank Learning and it enhanced on NSGA-II.

	MOEA/D			NSGA-II			PRL		
	W	D	L	W	D	L	W	D	L
SPEA2	5	0	14	15	4	0	19	0	0
MOEA/D				15	0	4	16	2	1
NSGA-II							18	1	0

(a) Generational Distance

	MOEA/D			NSGA-II			PRL		
	W	D	L	W	D	L	W	D	L
SPEA2	6	1	12	15	4	0	18	1	0
MOEA/D				12	3	4	15	1	3
NSGA-II							15	4	0

(b) Inverted Generational Distance

	MOEA/D			NSGA-II			PRL		
	W	D	L	W	D	L	W	D	L
SPEA2	4	6	7	9	10	0	13	6	0
MOEA/D				10	7	2	13	5	1
NSGA-II							12	7	0

(c) Hypervolume

every 500 evaluations starting from 1500 - 4000 evaluations on all three performance metrics. Overall, the PRL enhanced NSGA-II displays superiority over the classic NSGA-II on almost all three performance metrics. The few exceptions can be noted in Figure 3(b), i.e., the dark shade region, where the PRL enhanced NSGA-II under-performed the NSGA-II significantly on 2 test problems, namely UF1 and UF7, based on the Inverted Generational Distance metric. On Generational Distance and Hypervolume metrics, however, PRL enhanced

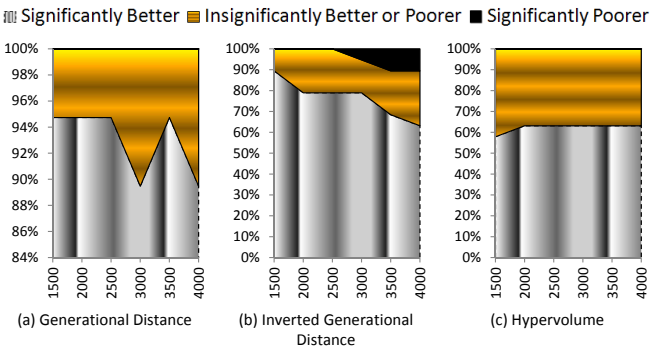

Fig. 3: Percentage of all the 19 test problems, for a limited computational budget of up to 4000 evaluations, where the PRL enhanced NSGA-II is significantly or insignificantly better or poorer than the classical NSGA-II. The x-axis denotes the number of "expensive" evaluations call to $\mathbf{f}(.)$.

NSGA-II remains to significantly outperform the NSGA-II on these two test problems. In what follows, we conducted further analysis on the converged non-dominated solutions of the algorithms for representative problems where the Pareto Rank Learning scheme has been successful in bringing about significant cost savings and another where the PRL enhanced NSGA-II under-performed the classical NSGA-II.

The detailed analysis on problem ZDT1 depicted in Figure 4 showcases the non-dominated solutions obtained by the PRL enhanced NSGA-II and classical NSGA-II algorithms for every 500 evaluations until the maximum budget of 4000 evaluations is reached. At each 500 evaluation intervals, the non-dominated solutions of PRL enhanced NSGA-II are noted to be consistently closer to the true Pareto front (TPF) than those of NSGA-II. In addition, the acceleration of non-dominated solutions in the PRL enhanced NSGA in converging to the TPF can be noted to surpass over the classical NSGA-II from 2000 evaluations. Eventually, the converged non-dominated solutions of PRL enhanced NSGA-II become inseparable from that of TPF at end of 4000 evaluations.

Figure 5, on the other hand, depicts the non-dominated solutions obtained of the UF7 test problem. It can be observed that the non-dominated solutions obtained by PRL enhanced NSGA-II appears to be closer to the true Pareto front than that of classical NSGA-II, i.e., majority of the non-dominated solutions of NSGA-II seem to be dominated by the solutions obtained via PRL enhanced NSGA-II, despite the Inverted Generational Distance metric indicating otherwise. In what follows, we analyze the reason for the seemingly contradicting observations. The Inverted Generational Distance metric gives the nearest distance between each of the non-dominated solutions of TPF and the converged non-dominated solutions of the respective algorithms. In subfigure 5(f), it can be observed that portion of the non-dominated solutions of the TPF (as denoted by the oval shape curve) are generally nearer to the non-dominated solutions of the classical NSGA-II, thus leading to the lower values in Inverted Generational Distance for NSGA-II.

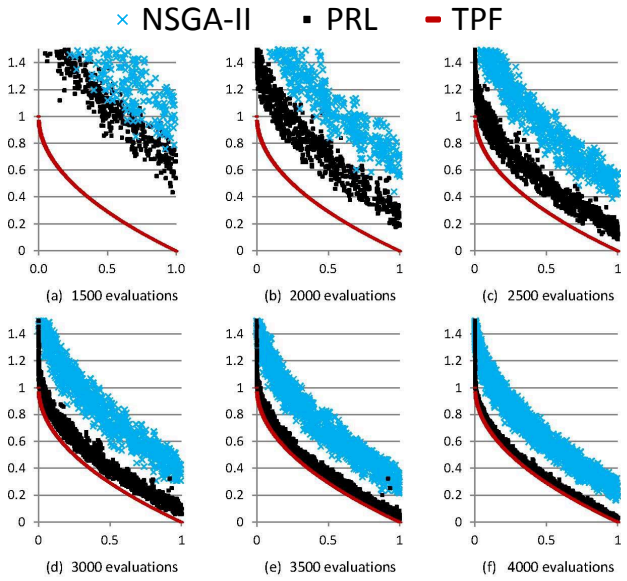

Fig. 4: The non-dominated solutions of NSGA-II and PRL enhanced NSGA-II for all 30 runs on ZDT1. TPF denotes the true Pareto front. The x-axis and y-axis represent the first and second objectives of ZDT1, respectively.

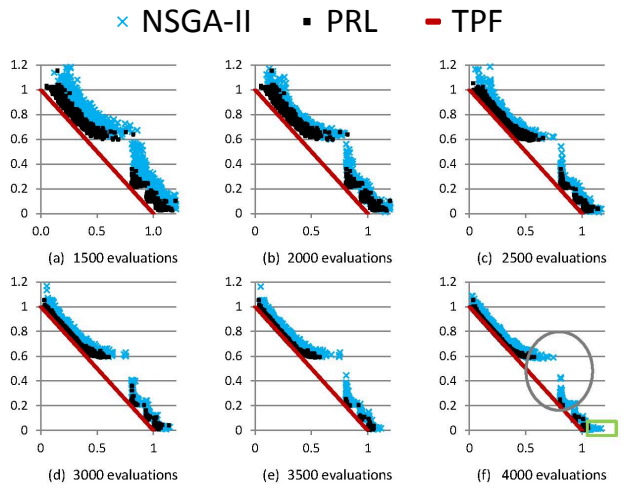

Fig. 5: The non-dominated solutions of NSGA-II and PRL enhanced NSGA-II for all 30 runs on UF7. TPF denotes the true Pareto front. The x-axis and y-axis represent the first and second objectives of UF7, respectively.

In addition, the non-dominated solutions of NSGA-II inside the oval bounding curve are not penalized by Inverted Generational Distance metric since there exists another non-dominated solution from NSGA-II that is nearer to the true Pareto solution of which the first objective has value one. Thus, it is a result of these biases in the Inverted Generational Distance metric that the PRL enhanced NSGA-II has been assessed to significantly under-performed the classical NSGA-II. Apparently, our detailed analysis on the UF1 problem also uncovered the same issues in the Inverted Generational Distance metric. Thus, the Inverted Generational Distance does not always serve as an appropriate indicator for accessing non-dominated solutions, particularly on those obtained on UF1 and UF7. In summary of this paper under limited expensive evaluations of 4000, the non-dominated solutions of the PRL enhanced NSGA-II are in overall nearer to the true Pareto front than that of NSGA-II.

Finally, we present in Figure 6 the converged non-dominated solutions obtained by the classical NSGA-II and PRL enhanced NSGA-II at 4000 evaluations for some of the other problems considered in the study, with Subfigures 6(a), 6(b, e and g) and 6(c, d, f and h) depicting the representative convex, concave and discontinuous test problems. From the figure, it can be observed that in overall the non-dominated solutions attained by PRL enhanced NSGA-II are closer to the true Pareto front than those of the classical NSGA-II. In particular, it can be clearly seen in Subfigures 6(e, f and h) that the non-dominated solutions of the PRL enhanced NSGA-II have converged close to the true Pareto front, whereas the non-dominated solutions obtained by NSGA-II remain quite some distances away.

IV. Conclusion and Future Work

In this paper, a Pareto Rank Learning scheme is proposed for enhancing multi-objective evolutionary optimization on problems that are either costly or hazardous to construct or extremely computationally intensive to compute. In contrast to the building of regression models for each separate objective in predicting the fitness values of the potential solutions in surrogate-assisted multi-objective evolutionary search, through learning from the solutions archived along the search the proposed Pareto Rank Learning scheme constructs surrogate models that predict the Pareto front ranks of the offspring for the multiple objectives simultaneously. By limiting the "expensive" evaluations to only the elite solutions, numerous numbers of unnecessary evaluations on the inferior solutions can be avoided, thus arriving at near the true Pareto front more efficiently. Experimental study on 19 multi-objective problems have also been made on various state-of-the-art algorithms, including the Pareto Rank Learning enhanced NSGA-II. Experimental results have been reported to show that the Pareto Rank Learning enhanced NSGA-II converged to the true Pareto front significantly faster than the classical NSGA-II on many of the test problems considered. Hence, in the manifest of limited resources, the Pareto Rank Learning scheme thus serves as an indispensable tool. This preliminary study of the Pareto Rank Learning in Pareto-Dominance MOEA thus opens up broad opportunities for future works on more general MOEAs.

Acknowledgment

This work is partially funded and supported by the Media Development Authority of Singapore, Singapore-MIT GAMBIT Game Lab and the Center for Computational Intelligence (C2I) at Nanyang Technological University.

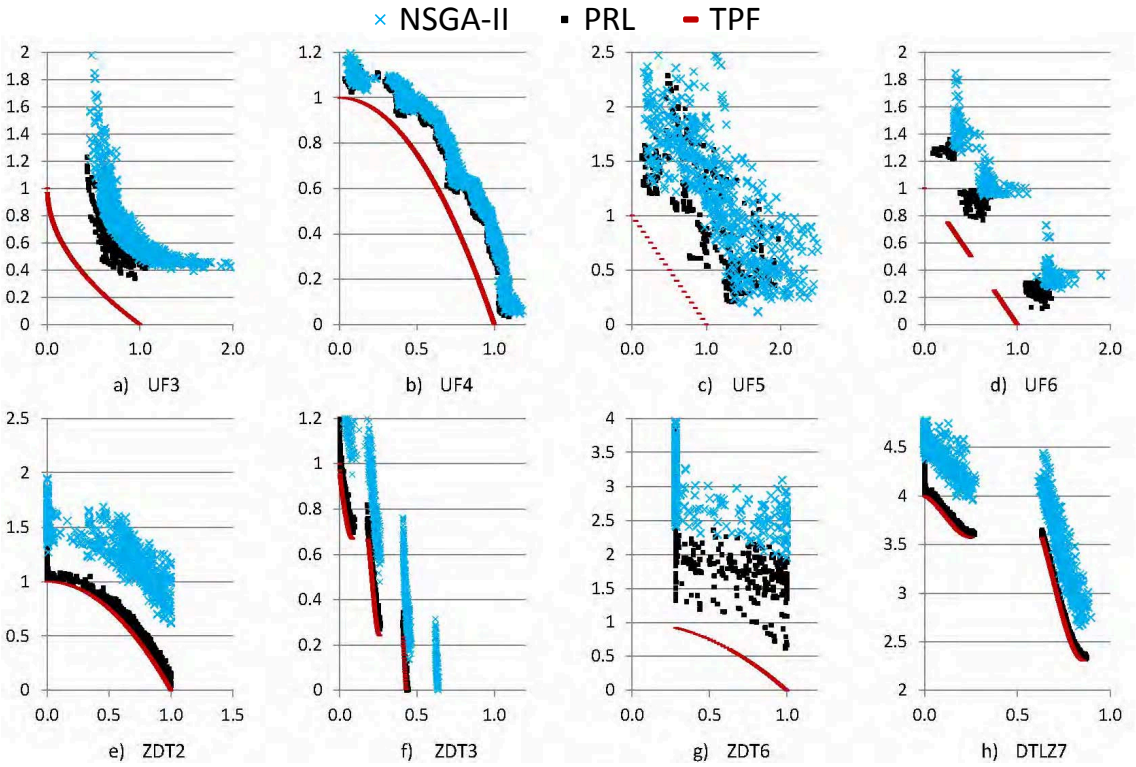

Fig. 6: The non-dominated solutions of NSGA-II and PRL enhanced NSGA-II for all 30 runs on the various problems at 4000 evaluations. TPF denotes the true Pareto front. The x-axis and y-axis denote the first and second objective of the optimization problem, respectively.

REFERENCES

[1] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, pp. 117–132, 2003.

[2] J. Durillo, A. Nebro, C. Coello, J. Garci anda Nieto, F. Luna, and E. Alba, "A study of multiobjective metaheuristics when solving parameter scalable problems," *IEEE Transactions on Evolutionary Computation*, vol. 14, pp. 618 –635, 2010.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[4] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," in *EUROGEN*, vol. 3242, no. 103, 2001, pp. 1–21.

[5] S. M. Lucas and G. Kendall, "Evolutionary Computation and Games," *IEEE on Computational Intelligence Magazine*, vol. 1, pp. 10–18, 2006.

[6] D. Lim, Y. Jin, Y. S. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *IEEE Transactions on Evolutionary Computation*, vol. 14, pp. 329–355, 2010.

[7] S. D. Handoko, C. K. Kwoh, and Y.-S. Ong, "Feasibility structure modeling: An effective chaperone for constrained memetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 14, pp. 740–758, 2010.

[8] M. G. C. Tapia and C. A. C. Coello, "Applications of multi-objective evolutionary algorithms in economics and finance: A survey." in *IEEE Congress on Evolutionary Computation*, 2007, pp. 532–539.

[9] Y. Jin, M. Olhofer, and B. Sendhoff, "A framework for evolutionary optimization with approximate fitness functions," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 481 – 494, 2002.

[10] Y. S. Ong, P. B. Nair, and A. J. Keane, "Evolutionary Optimization of Computationally Expensive Problems via Surrogate Modeling," *American Institute of Aeronautics and Astronautics Journal*, vol. 41, pp. 687–696, 2003.

[11] Z. Zhou, Y. S. Ong, M. H. Lim, and B. S. Lee, "Memetic algorithm using multi-surrogates for computationally expensive optimization problems," *Soft Computing*, vol. 11, no. 10, pp. 957–971, 2007.

[12] M. N. Le, Y. S. Ong, and Q. H. Nguyen, "Optinformatics for schema analysis of binary genetic algorithms," in *Genetic and Evolutionary Computation*, 2008.

[13] Y.-S. Ong, M.-H. Lim, and X. Chen, "Memetic computation - past, present & future [research frontier]," *IEEE on Computational Intelligence Magazine*, vol. 5, pp. 24–31, 2010.

[14] X. Chen, Y.-S. Ong, M.-H. Lim, and K. C. Tan, "A multi-facet survey on memetic computation," *IEEE Transactions on Evolutionary Computation*, vol. 15, pp. 591 –607, 2011.

[15] J. Knowles, "Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, pp. 50 – 66, 2006.

[16] M. Emmerich, K. Giannakoglou, and B. Naujoks, "Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels," *IEEE Transactions on Evolutionary Computation*, vol. 10, pp. 421 –439, 2006.

[17] L. Li and H.-T. Lin, "Ordinal regression by extended binary classification," in *Neural Information Processing Systems*, 2007.

[18] W. Chu and S. S. Keerthi, "Support vector ordinal regression," *Neural Computation*, vol. 19, p. 2007, 2007.

[19] J. S. Cardoso and J. F. Pinto da Costa, "Learning to classify ordinal data: The data replication method," *Journal of Machine Learning Research*, vol. 8, pp. 1393–1429, 2007.

[20] A. Shashua and A. Levin, "Ranking with large margin principle: Two approaches," in *Neural Information Processing Systems*, 2003.

[21] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.

[22] J. Durillo, A. Nebro, and E. Alba, "The jmetal framework for multi-objective optimization: Design and architecture," *IEEE Congress on Evolutionary Computation*, pp. 1–8, 2010.