**Journal of the Operational Research Society (2011) 62**, 348–359 

www.palgrave-journals.com/jors/

# Hybrid approach for Pareto front expansion in heuristics

H Yapicioglu[1]*, H Liu[2], AE Smith[2] and G Dozier[3]

*[1]Anadolu University, Eskisehir, Turkey; [2]Auburn University, Auburn, USA; and [3]North Carolina A&T University, Greensboro, USA*

Heuristic search can be an effective multi-objective optimization tool; however, the required frequent function evaluations can exhaust computational sources. This paper explores using a hybrid approach with statistical interpolation methods to expand optimal solutions obtained by multiple criteria heuristic search. The goal is to significantly increase the number of Pareto optimal solutions while limiting computational effort. The interpolation approaches studied are kriging and general regression neural networks. This paper develops a hybrid methodology combining an interpolator with a heuristic, and examines performance on several non-linear bi-objective example problems. Computational experience shows this approach successfully expands and enriches the Pareto fronts of multi-objective optimization problems.

*Journal of the Operational Research Society* (2011) **62,** 348–359. doi:10.1057/jors.2010.151
Published online 27 October 2010

**Keywords:** kriging; general regression neural network; multi-objective optimization; heuristic search; hybrid methods



## 1. Introduction

Traditionally, multi-objective optimization problems (MOPs) have been solved by mathematical programming techniques. There are over 30 mathematical programming techniques addressing MOPs (Miettinen, 1999). However, most of these techniques are sensitive to the shape of the Pareto front and may not work well when the Pareto front is concave or disconnected. Since 1984 (Schaffer, 1984), population-based heuristics have been used to solve MOPs. In 1989, Goldberg *et al* (1989) proposed the Multi-objective Evolutionary Algorithm framework based on the idea of Pareto dominance. From there, many researchers began using the idea of Pareto dominance in conjunction with evolutionary algorithms (Deb, 2001; Coello Coello *et al*, 2007). With a set of alternative solutions, that is the population, evolutionary algorithms can find multiple members of a Pareto optimal set in a single run, and are less susceptible to the shape or continuity of the Pareto front than gradient-based methods. However, the non-dominated solutions are identified one by one and these heuristics provide less information on the contiguity of the Pareto front. This paper uses two non-linear statistical interpolators, kriging and the General Regression Neural Network (GRNN), to expand the Pareto front discovered by heuristic search.

The paper is organized as follows. In the second section, kriging is introduced. In the third section, the definition of GRNN is provided. The fourth section introduces concepts related to MOPs and presents the computational experience using six test functions. Finally, the paper concludes and provides future research directions.

*Correspondence: *H Yapicioglu, Department of Industrial Engineering, Anadolu University, Iki Eylul Kampusu Eskisehir, Turkey*
E-mail: hyapicio@anadolu.edu.tr

## 2. Kriging methodology

In the 1950s Krige first applied the technique to the estimation of gold deposits (Krige, 1951; Cressie, 1993). In the early 1960s Matheron (1973) further developed the mathematical theory of kriging, which is a statistical interpolator that calculates the best linear unbiased estimator of various functions of spatial processes (Cressie, 1993). Kriging has been used in applications ranging from contouring of maps, engineering design, and simulation meta-modelling (Olea, 1974; Sacks *et al*, 1989; Simpson *et al*, 2001; Kleijnen and van Beers, 2005).

### *2.1. The basic of kriging models*

Consider a stochastic process with the form below:

$$Z(\mathbf{X}) = \mu(\mathbf{X}) + \varepsilon(\mathbf{X}) \tag{1}$$

where $\mathbf{X}$ is a position vector, $\mu(\mathbf{X})$ is the mean term or a drift function showing the average behaviour of $Z(\mathbf{X})$, and $\varepsilon(\mathbf{X})$ is an error term with $E[\varepsilon(\mathbf{X})]=0$. Suppose the $n$ observed values are $Z(\mathbf{X}_1)$, $Z(\mathbf{X}_2)$, $\ldots$, $Z(\mathbf{X}_n)$. Kriging uses a linear combination of the observed values to estimate the process value at an unobserved point $\mathbf{X}$.

$$\hat{Z}(\mathbf{X}) = \sum_{i=1}^{n} \lambda_i Z(\mathbf{X}_i) \tag{2}$$

where coefficients $\lambda_i$ are chosen to minimize the variance subject to an unbiasedness condition. Without loss of generality, we assume that the drift function $\mu(\mathbf{X})$ consists of $M$ basis functions $y_m(\mathbf{X})$, $m=1, 2, \ldots, M$. Then, the unbiasedness condition becomes

$$y_l(\mathbf{X}) = \sum_{i=1}^{n} \lambda_i y_m(\mathbf{X}_i), \qquad m = 1, 2, \ldots, M \tag{3}$$

Thus minimizing the variance of estimation error becomes a constrained optimization problem that can be expressed as:

$$\begin{aligned} \min \ Var\big[\hat{Z}(\mathbf{X}) - Z(\mathbf{X})\big] = & \sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j Cov\big[Z(\mathbf{X}_i), Z(\mathbf{X}_j)\big] \\ & - 2\sum_{i=1}^{n} \lambda_i Cov[Z(\mathbf{X}_i), Z(\mathbf{X})] \\ & + Var[Z(\mathbf{X})] \end{aligned} \tag{4}$$

s.t.

$$y_m(\mathbf{X}) = \sum_{i=1}^{n} \lambda_i y_m(\mathbf{X}_i), \qquad m = 1, 2, \ldots, M \tag{5}$$

Kriging transforms the model above into an unconstrained optimal problem by introducing Lagrangian multipliers $u_m$ where $m=1, 2, \ldots, M$. By using partial differentiation with respect to $\lambda_i (i=1, 2, \ldots, n)$ and $u_m (m=1, 2, \ldots, M)$, we obtain the kriging equation systems, which are expressed in the matrix form as follows.

$$\begin{bmatrix} c_{11} & \cdots & c_{1n} & y_1(\mathbf{X}_1) & \cdots & y_1(\mathbf{X}_n) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ c_{n1} & \cdots & c_{nn} & y_M(\mathbf{X}_1) & \cdots & y_M(\mathbf{X}_n) \\ y_1(\mathbf{X}_1) & \cdots & y_1(\mathbf{X}_n) & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ y_M(\mathbf{X}_1) & \cdots & y_M(\mathbf{X}_n) & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_n \\ u_1 \\ \vdots \\ u_M \end{bmatrix} = \begin{bmatrix} c_1 \\ \vdots \\ c_n \\ y_1(\mathbf{X}) \\ \vdots \\ y_M(\mathbf{X}) \end{bmatrix} \tag{6}$$

where $c_{ij}$ represents $Cov(Z(\mathbf{X}_i), Z(\mathbf{X}_j))$, which is the covariance between the sample points $\mathbf{X}_i$ and $\mathbf{X}_j$, and $c_i$ represents the covariance $Cov(Z(\mathbf{X}_i), Z(\mathbf{X}))$. Equation (6) is called the kriging matrix. When $\mu(\mathbf{X})$ is equal to zero, a non-zero constant, or an unknown polynomial, the corresponding kriging models are termed simple, ordinary, and universal kriging, respectively.

### *2.2. Covariance functions of kriging models*

In the kriging matrix, the covariance $c_{ij}$ is generally calculated by using a covariance function. We consider three simple common functions (Trochu, 1993), shown below, where $h$ denotes the distance between two points and $d$ denotes the distance over which the covariance between two points can be assumed null. Hence, selection of $d$ is problem specific but the method is not too sensitive to $d$.

1. The pure nugget effect covariance function:

$$c(h) = \begin{cases} 1, & \text{if } h = 0 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

2. The linear covariance function:

$$c(h) = \begin{cases} 1 - h/d, & \text{if } h < d \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

3. The cubic covariance function:

$$c(h) = \begin{cases} 1 - 3(h/d)^2 + 2(h/d)^3, & \text{if } h < d \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

## 3. The general regression neural network

General regression neural networks (GRNNs) were developed by Specht (1991). GRNN can be viewed as an extension to $k$-nearest neighbour approximation using a distance-weighted average and a global neighbourhood. GRNNs are 'lazy' learners (Mitchell, 1997) since they utilize all training instances and do not form a model of the training set. A simple GRNN is defined as follows. There is a set of $n$ training elements $\{(b_1, c_1), (b_2, c_2), \ldots, (b_n, c_n)\}$, where $c_i$ is the correct output of the $i$th training instance $b_i$. In GRNN, each training element defines a neuron. Let the classification of some input instance, $x_1$, be

$$s_1 = \sum_{i=1}^{p} c_i h_i (x_1 - b_i) \tag{10}$$

$$s_2 = \sum_{i=1}^{p} h_i (x_1 - b_i) \tag{11}$$

$$x_2 = s_1/s_2$$

where

$$h_i(x_1, b_i) = \exp(-(\|x_1 - b_i\|^2)/2\sigma_i^2. \tag{12}$$

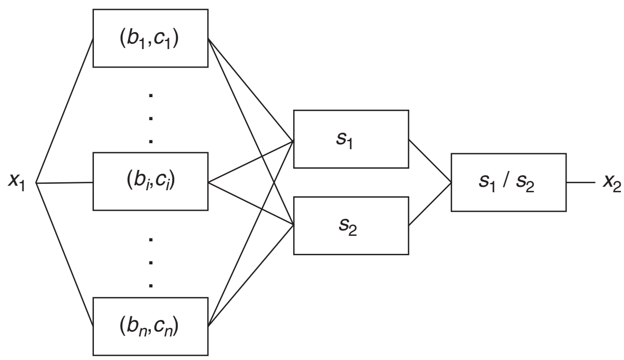


**Figure 1** GRNN architecture.

The function, $h_i$, is referred to as the $i$th hidden function. The width of the hidden function is denoted as $\sigma$ and defines the influence domain of the $i$th neuron on the output $x_2$. Figure 1 provides a GRNN architecture. As can be seen, GRNN does not require a special training phase; each training instance serves as a neuron.

GRNN has been used in various applications since its inception. Examples include pattern recognition (Polat and Yıldırım, 2008; Wu and Ye, 2009) and estimation/prediction (Werner and Obach, 2001; Casey *et al*, 2006). In an earlier work, Yapicioglu *et al* (2006) show how GRNN can be used to enhance results obtained from a multi-objective optimizer for problems involving two decision variables. Later Garrett *et al* (2007) extended this to three or more decision variables.

## 4. Expansion of Pareto optimal set of multi-objective optimization

In this section, we provide the general framework for obtaining the candidate non-dominated solutions to a bi-objective two variable non-linear optimization problem. The method could be expanded to three or more objectives straightforwardly, only requiring adding the additional objective functions to the computation routine. This would, however, increase the computational effort.

### *4.1. MOPs and initial solutions*

In an MOP, the aim is to simultaneously optimize a set of objective functions subject to a set of constraints. Expressed mathematically:

$$\begin{aligned} \text{min/max} \quad & f_t(\mathbf{X}), && t = 1, 2, \ldots, T \\ \text{s.t.} \quad & g_j(\mathbf{X}) \geqslant 0, && j = 1, 2, \ldots, J \\ & h_k(\mathbf{X}) = 0, && k = 1, 2, \ldots, K \\ & x_i^{(L)} \leqslant x_i \leqslant x_i^{(U)}, && i = 1, 2, \ldots, n \end{aligned} \tag{13}$$

where $T$ is the number of objective functions, subject to $J$ inequality and $K$ equality constraints and $n$ decision variables denoted by $\mathbf{x}$ are bounded from below and above by $\mathbf{x}^{(L)}$ and $\mathbf{x}^{(U)}$, respectively (Deb, 2001). From here on, we will refer to the set of objective functions as $\mathbf{f}$. In this study, we consider MOPs where $T=2$, $n=2$, and $J=K=0$.

In MOPs the objective functions are mathematical descriptions of performance criteria that often conflict with each other. In the presence of more than one objective function, the notion of 'optimum' is originally proposed by Edgeworth in 1881 and later generalized by Pareto in 1896 (Miettinen, 1999). According to this definition, a solution is Pareto optimal if there does not exist any other solution that would improve the value of some criteria without causing a simultaneous deterioration in at least one other criterion. This defines a set of solutions called the Pareto optimal set or the set of non-dominated solutions. The image of the Pareto optimal set in the solution space is called the Pareto front.

### *4.2. Multi-objective particle swarm optimization*

After its inception in 1995 (Kennedy and Eberhart, 1995), particle swarm optimization (PSO) received attention in the evolutionary computation community due to its speed of convergence and ease of use. In 2002, the first studies applying PSO in multi-objective problems appeared (Coello Coello and Lechuga, 2002; Fieldsend and Singh, 2002). Since then, many different variants of PSO in multi-objective settings have been proposed. A good survey on multi-objective PSOs can be found in Reyes-Sierra and Coello Coello (2006). Following the survey paper of Reyes-Sierra and Coello Coello (2006), papers by Yapicioglu *et al* (2007) and Agrawal *et al* (2008) are among the more recent additions to the multi-objective PSO literature.

In this paper, we use a bi-objective particle swarm optimization (bi-PSO) (Yapicioglu *et al* 2007) to obtain an initial Pareto set of solutions, but any heuristic could be used, including genetic algorithms, ant colony systems, and evolutionary strategies. In Yapicioglu *et al*'s bi-PSO the set of non-dominated solutions is kept in an external repository during search. During particle update, a randomly selected member of the external Pareto repository is selected as the neighbourhood best, so that every member of the Pareto front has an equal probability of contributing to the search direction. Whenever a particle finds a new solution that dominates its incumbent 'best', it replaces the incumbent and is added to the external Pareto repository. After all particles are updated and evaluated, the external Pareto repository is updated by using a dominance control routine to remove any dominated solutions. The steps of the bi-PSO from Yapicioglu *et al* (2007) are provided below.

*Step* 1: $t=0$; Initialize Swarm($t$);

Evaluate the particles of the Swarm(t);

Initialize Pareto_Front ($t$);//Repository at cycle $t$ (Pareto_Front($t$) is empty at $t=0$)
*Step* 2: While (Not Done){
for each particle $i \in$ Swarm($t$) {
Select a member from Pareto_Front($t$) randomly;
Update ($i$, Pareto_Front($t$));
Evaluate particle $i$;
}
Update Pareto_Front($t$);
$t := t+1$;
}
*Step* 3: Print out Pareto_Front($t$), end.

### *4.3. Expansion of Pareto optimal sets using kriging*

In this section, kriging is used to expand a Pareto optimal set obtained from a heuristic. With decision variables $(x_1, x_2)$ we arbitrarily choose $x_1$ to serve as the input, and $x_2$ is treated as the output to construct a kriging model. The steps are defined below.

*Step* 1: Define the set of objective functions $\mathbf{f}=(f_1(\mathbf{x}), f_2(\mathbf{x}))$ as a function of decision variables, $\mathbf{x}=(x_1, x_2)$, and let FS be an empty solution set of $(\mathbf{x}, \mathbf{f})$;
*Step* 2: Let $ND_F$ be the initial non-dominated solution set obtained from the heuristic and construct a kriging model using $ND_F$;
*Step* 3: Divide the problem-defined interval $(x_1^{(L)}, x_1^{(U)})$ of $x_1$ into $E$ equi-spaced intervals, and perform the following loop:

for($i=0$; $i \leqslant E$; $i++$){
$x_{1\,i} := (x_1^{(L)} + (x_1^{(U)} - x_1^{(L)})i/E)$;
compute $x_{2i}$ by using the kriging model to obtain $\mathbf{x}_i$;
compute $\mathbf{f}_i = \mathbf{f}(\mathbf{x}_i)$;
FS: $= (\mathbf{x}_i, \mathbf{f}_i) \bigcup$ FS;
}end for

*Step* 4: Set the candidate non-dominated solution set $CNS' = FS \bigcup (ND_F)$;
*Step* 5: Obtain the non-dominated solution set $NS$ by performing a Pareto optimality check of $CNS'$.

In other words, we start by constructing a kriging model based on the non-dominated solutions obtained from the bi-PSO (Step 2). Next, we assign arbitrary values to one of the decision variables and calculate the other decision variable using the kriging model. The objective function values for each candidate non-dominated solution are computed using the objective function (Step 3). At this point, we do not know whether a particular solution obtained through kriging is actually a non-dominated solution. Next, we combine the non-dominated solutions from the bi-PSO and the candidate non-dominated solutions obtained in Step 3 (Step 4) into a single set ($CNS'$). Finally, we prune $CNS'$ by performing a Pareto optimality check (Step 5). For this, we use the dominance check routine explained in Deb (2001).

### *4.4. Expansion of Pareto optimal sets using GRNN*

To use GRNN for the Pareto front expansion, the method is similar to that of kriging in Section 4.3. The explanation at the end of Section 4.3 is also applicable to this method, except that we use a GRNN instead of kriging as the expansion tool.

*Step* 1: Define the set of objective functions $\mathbf{f}=(f_1, f_2)$ as a function of decision variables, $\mathbf{x}=(x_1, x_2)$, and let FS be an empty solution set of $(\mathbf{x}, \mathbf{f})$;
*Step* 2: Define the initial non-dominated solution set obtained from the heuristic as $ND_F$, and construct a GRNN using $ND_F$;
*Step* 3: Divide the problem-defined interval $(x_1^{(L)}, x_1^{(U)})$ of $x_1$ into $E$ equi-spaced intervals, and perform the following loop:

for($i=0$; $i \leqslant E$; $i++$){
$x_{1i} := (x_1^{(L)} + (x_1^{(U)} - x_1^{(L)})i/E)$;
compute $x_{2i}$ by using the GRNN to obtain $\mathbf{x}_i$;
compute $\mathbf{f}_i = \mathbf{f}(\mathbf{x}_i)$;
FS: $= (\mathbf{x}_i, \mathbf{f}_i) \bigcup$ FS;
}end for

*Step* 4: Set the candidate non-dominated solution set $CNS' = FS \bigcup (ND_F)$;
*Step* 5: Obtain the non-dominated solution set $NS$ by performing Pareto optimality check to $CNS'$.

Figure 2 shows the hybrid method of expansion of Pareto optimal sets.

## 5. Computational experience

We experimented with six test problems that were adopted from Yapicioglu *et al* (2006). All test problems are listed in Appendix A. The determinant of the influence domain of the neurons in GRNN, $\sigma$, is set to 0.3 throughout experimentation. Although the method is not very sensitive to this value, when the value of $\sigma$ is decreased, the quantity of produced non-dominated solutions decreases. When the value of $\sigma$ increased, the GRNN is able to produce a larger number of candidate non-dominated solutions. However, at Step 5 of the algorithm the number of eliminated candidate non-dominated solutions decreases due to the inferior quality of the estimates.

### *5.1. Kriging parameter setting using Problem 1*

First, we analysed the effects of covariance functions and influence distances of covariance, $d$ in Equations (7)–(9), on

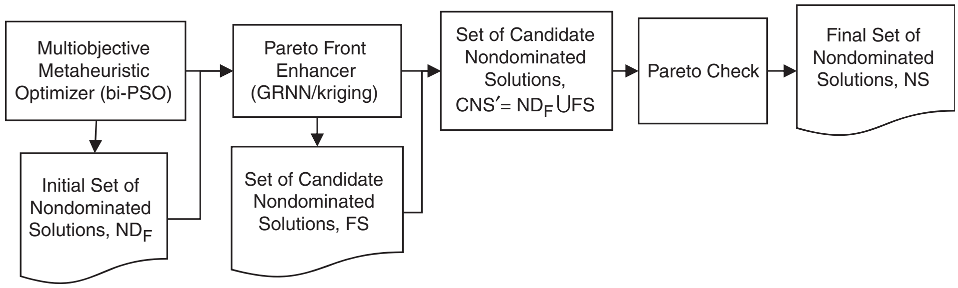


**Figure 2** Hybrid method of expansion of Pareto optimal sets.

**Table 1** Expanded solutions of different kriging models with different covariance functions and range of covariance existence

| | | | *Influence distance of covariance (SD)* | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Kriging* | *Covariance function* | *PSO* | *0 (Nugget)* | | 0.5 | | 1 | | 2 | | 3 | | 4 | | 5 | |
| | | | *NKS* | *PDKS* | *NKS* | *PDKS* | *NKS* | *PDKS* | *NKS* | *PDKS* | *NKS* | *PDKS* | *NKS* | *PDKS* | *NKS* | *PDKS* |
| Simple | Linear | 185 | 2397 | 4 | 2926 | 68 | 2580 | 63 | 4204 | 49 | 6729 | 46 | 7085 | 46 | 7085 | 46 |
| | Cubic | 185 | 2397 | 4 | 4944 | 49 | 5551 | 64 | 3600 | 62 | 3413 | 67 | 2297 | 72 | 2131 | 75 |
| Ordinary | Linear | 185 | 4078 | 10 | 2953 | 65 | 2582 | 65 | 4220 | 49 | 6784 | 46 | 7085 | 46 | 7085 | 46 |
| | Cubic | 185 | 4078 | 10 | 4945 | 50 | 5498 | 59 | 3644 | 64 | 3352 | 67 | 2529 | 74 | 2093 | 77 |

NKS: Non-dominated kriging Solutions; PDKS: PSO Solutions Dominated by kriging Solutions.

the performance of different kriging models. The nugget, linear, and cubic covariance functions with seven different influence distances of covariance are investigated for Problem 1. For the other five problems, the sample standard deviation is used as the measure unit of influence distance of covariance. The advantage of choosing the sample standard deviation is that it is easy to calculate and allows scaling among problems. To establish the kriging models for Problem 1, all 185 non-dominated solutions obtained by the bi-PSO were used as the initial set.

The results of the experiments are summarized in Table 1. For simple kriging, when we use the influence distance of covariance of five *SDs* (eg, $d = 5$ *SD*), 7085 new Pareto solutions are generated. As stated previously, as $d$ increases, more points are included in the estimation process and more new Pareto solutions are identified. The number of the new Pareto solutions is 38.3 times the initial number of Pareto solutions obtained by the PSO. Forty-six solutions of the initial non-dominated set were dominated by the new solutions obtained by kriging. The combined set (after Step 5) has 7224 non-dominated solutions.

In order to assess the impact of the type of kriging (Simple *versus* Ordinary) and the covariance distance, we performed an analysis of variance (ANOVA), results of which are summarized in Table 2 and Table 3, for linear and cubic covariance functions, respectively. As can be verified from the ANOVA results, the effect of influence distance of covariance is significant. However, the type of kriging model is not significant.

**Table 2** ANOVA of linear covariance function

| *Source* | *df* | *SS* | *MS* | *F* | *P* |
|---|---|---|---|---|---|
| Kriging | 1 | 226 569 | 226 569 | 1.33 | 0.29 |
| Sigma | 5 | 50 860 710 | 10 172 142 | 59.92 | 0.00 |
| Error | 7 | 1 188 319 | 169 760 | | |
| Total | 13 | 52 275 597 | | | |

*df:* degrees of freedom, *SS:* Sum of Squares, *MS:* Mean Squared Error, *F*: Corresponding F statistics, *p:* probability of rejection.

**Table 3** ANOVA of cubic covariance function

| *Source* | *DF* | *SS* | *MS* | *F* | *P* |
|---|---|---|---|---|---|
| Kriging | 1 | 232 974 | 232 974 | 1.15 | 0.32 |
| Sigma | 5 | 18 627 010 | 3 725 402 | 15.37 | 0.00 |
| Error | 7 | 1 211 774 | 173 111 | | |
| Total | 13 | 20 071 758 | | | |

*df:* degrees of freedom, *SS:* Sum of Squares, *MS:* Mean Squared Error, *F*: Corresponding F statistics, *p*: probability of rejection

### *5.2. Test problems comparing kriging and GRNN*

Since our experimentation on Problem 1 showed that the type of kriging model is not important we chose the simplest of the two, ordinary kriging, to be used with a linear covariance function with $SD = 5$ (Table 4) with two exceptions. The first exception is that we increased the influence distance of covariance to nine for Problem 2 as for this problem the initial Pareto Front is sparser compared to the remaining five test problems. For Problem 5, for both linear and cubic covariance functions the kriging matrix is singular and its inverse does not exist. Therefore we use the nugget covariance with $SD = 0$. In general, when the influence distance of covariance increases, more points will be included in the estimation process, and thus the performance of kriging will be better.

Table 5 shows the performance of the kriging models compared with that of the GRNN (Yapicioglu *et al*, 2006). For Problem 1 in Table 5, we can see that when 10 000 and 9815 $x_1$'s were respectively fed to the kriging models and the GRNN, they generated 7231 and 5230 new Pareto optimal solutions, respectively. In the fifth column of Table 5, the numbers of Pareto optimal solutions obtained by the PSO and dominated by the solutions generated by kriging or GRNN are 46 and 35, respectively. For the six test problems, the numbers of non-dominated solutions obtained by kriging are from about 13.5 to 39.1 times larger and the average is 29.4 times larger (the column termed 'Yield'), and for the GRNN, the numbers are from 17.5 to 29.3 times larger with the average being 26.3 times larger. These results are true except for Problem 3 where the performance of kriging is not as good. The reason is that we can only use the nugget covariance function that negatively impacts the number of new Pareto solutions found.

Table 6 lists the intervals of the objective functions. The intervals of the objective function values obtained by the kriging and GRNN models are quite close to, but slightly wider, than the intervals of the objective function values obtained by the PSO.

Results are shown graphically in Appendix B. The graphs show that in all test problems, kriging and GRNN

**Table 4** Parameter settings of kriging models for different problems

| *Problem* | *Kriging type* | *Covariance function* | *Influence distance of covariance (SD)* |
|---|---|---|---|
| 1 | Ordinary | Linear | 5 |
| 2 | Ordinary | Linear | 9 |
| 3 | Ordinary | Linear | 5 |
| 4 | Ordinary | Linear | 5 |
| 5 | Ordinary | Nugget | 0 |
| 6 | Ordinary | Linear | 5 |

**Table 5** Performance overview

| *Problem* | *PSO* | *Kriging* | | | | *GRNN* | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | *Input* | *NKS* | *PSDKS* | *Yield* | *Input* | *NGS* | *PSDGS* | *Yield* |
| 1 | 185 | 10 000 | 7224 | 46 | 39.1 | 9815 | 5230 | 35 | 28.3 |
| 2 | 143 | 10 000 | 3793 | 55 | 26.5 | 9857 | 3763 | 42 | 27.3 |
| 3 | 94 | 10 000 | 2027 | 26 | 21.6 | 9906 | 1715 | 28 | 20.4 |
| 4 | 178 | 10 000 | 6717 | 50 | 37.7 | 9822 | 5213 | 59 | 29.3 |
| 5 | 142 | 10 000 | 1912 | 1 | 13.5 | 9858 | 2474 | 13 | 17.5 |
| 6 | 128 | 10 000 | 4765 | 64 | 37.2 | 9872 | 4487 | 39 | 35.1 |
| | | | | Average = 29.3 | | | | Average = 26.3 | |

NKS: Non-dominated kriging Solutions; NGS: Non-dominated GRNN Solutions; PSDKS: PSO Solutions Dominated by kriging Solutions; PSDGS: PSO Solutions Dominated by GRNN Solutions.

**Table 6** Extreme values of the objective function values

| *Problem* | *Objective function* | *PSO* | | *Kriging* | | *GRNN* | |
|---|---|---|---|---|---|---|---|
| | | *Min* | *Max* | *Min* | *Max* | *Min* | *Max* |
| | 1 | 240.53 | 1212.82 | 240.52 | 1246.1 | 240.53 | 1279.5 |
| 1 | 2 | 0.17 | 0.91 | 0.16 | 0.92 | 0.16 | 0.91 |
| | 1 | 240.53 | 1029.08 | 240.53 | 1027.32 | 240.53 | 1040.81 |
| 2 | 2 | 0 | 1362.72 | 0 | 1362.72 | 0 | 1362.69 |
| | 1 | 308.61 | 1034.65 | 308.79 | 1033.08 | 308.61 | 1062.46 |
| 3 | 2 | 0 | 1325.94 | 0 | 1325.94 | 0 | 1325.94 |
| | 1 | 35.96 | 124.04 | 35.95 | 124.93 | 35.96 | 125.53 |
| 4 | 2 | 0.78 | 9.78 | 0.78 | 9.85 | 0.78 | 9.85 |
| | 1 | 41 | 122 | 41.19 | 42 | 41 | 122 |
| 5 | 2 | 2 | 8 | 2.19 | 3 | 2 | 8 |
| | 1 | 271.88 | 3064.21 | 272.79 | 3331.38 | 267.06 | 3607.31 |
| 6 | 2 | 32 711 574 | 50 975 833 | 32 687 930 | 50 923 713 | 32 704 400 | 51 270 600 |

expand the initial Pareto front through both interpolation and extrapolation.

### *5.3. Discussion of kriging versus GRNN*

*Computational Complexity*: The algorithm complexity of kriging is dominated by calculating the inverse of the kriging matrix. The typical algorithm in linear algebra to obtain the inverse of a matrix is Gaussian elimination or Gauss–Jordan elimination. Even if we use the best algorithms such as the Strassen (1969) algorithm $O(n^{2.807})$ or Coppersmith and Winograd (1990) algorithm $O(n^{2.376})$ for the matrix inversion, the computational complexity of the kriging method remains costly. However, it is not necessary for kriging to use all Pareto efficient solutions from the heuristic to estimate the values of unknown points. The covariance between an estimated point and observation points far away is very small, and therefore we could use only points close to the estimated point. In addition, since nearby points have strong covariance, we could choose one or several representative points from a cluster of neighbouring points, because it is fairly redundant to use all of the observation values in a neighbourhood. Employing these two reduction mechanisms, the size of the kriging matrix can be limited no matter how large $n$ is.

The complexity of GRNN is order $O(n)$, the number of initial non-dominated solutions that are used to construct the GRNN. This is because there is no iterative training phase and the GRNN training method is a one pass algorithm with one iteration per initial solution. Neither of these techniques adds complexity to the bi-PSO algorithm, as they both are initiated after the bi-PSO terminates. By themselves the GRNN and kriging methods each take around 10 CPU seconds to run with the implementations used in this paper on a PC with Pentium Dual-Core CPU running at 2.60 GHz, and 3 GB of RAM. The bi-PSO used in this paper takes around 2 min to run on the same computer with the same test cases. Therefore, compared with the PSO either the GRNN or the kriging are minor computational expenses. Increasing the allotted time for the bi-PSO for another 10 CPU seconds or so would not have an equal effect of employing GRNN or kriging in terms of the number of non-dominated solutions discovered.

*Parameter setting*: Neither kriging nor GRNN has many parameters and both are fairly robust to parameter settings. For simple and ordinary kriging, we only need to choose a covariance function and influence distance of covariance. There are not many choices for these and often the problem data dictate or indicate a choice. The value of $\sigma$ in GRNN needs be chosen either by experimentation or through an optimization procedure.

*Estimated values and their accuracy*: Interpolation from kriging can expand the range of the known observation values. In addition, if an estimated point is one of the observation points, the estimated value of that point is equal to its observation value (kriging is an exact interpolator). In addition to the estimated values, kriging can provide the variance of errors of estimated and actual values, which can describe the accuracy of estimated values. GRNN can also expand the range of known observation values. However, it is very difficult to know the variance of errors between the estimated and actual values.

*Model assumptions*: Neither method has statistical distribution assumptions other than the one of the appropriateness of the sample used to construct the model. Therefore, either could be used under a wide variety of conditions without knowing or specifying certain forms or properties of the input/output relationship to achieve interpolation.



## 6. Conclusions and future work

This paper described a novel hybrid system of a heuristic multiple objective optimizer with a statistical estimator to expand the number of Pareto optimal solutions. We compare the performances of two different expanders, GRNN and kriging, and both methods are successful estimators of non-dominated solutions. Both found a wide range of solutions that both interpolated and extrapolated the Pareto front found by the PSO. In our six test problems, the numbers of non-dominated solutions obtained by kriging are from 13.5 to 39.1 times higher than the initial input number of non-dominated solutions. For the GRNN, the numbers are from 17.5 to 29.3 times higher. This can offer real value when a decision maker desires a wide range of potential solutions and/or wants to identify the form of the Pareto set or the Pareto front more fully. In addition, although the tested six problems have two objective functions, both GRNN and kriging can accommodate more than a single input that would allow a straightforward extension to three or more objectives. The effort to build either a kriging or GRNN model is not significant. We showed that neither is too sensitive to the few parameters that need to be set, mainly the width parameter of the GRNN ($\sigma$) and the width parameter of kriging (covariance influence distance, *SD*).

Of course, both have some drawbacks. Each represents an additional computational step in the hybrid method, though the computational complexity of each is manageable. Another drawback occurs when the inverse of kriging matrix does not exist and the nugget covariance function must be used. A third consideration is in the design and training of the GRNN network. In the current study, we do not use an iterative training phase. Thus, we had to include all of the non-dominated solutions from the bi-PSO. However, the GRNN might have been pruned had an iterative training phase employed. A leaner GRNN can generate the enhanced non-dominated set faster than a more crowded one. Another aspect that must be considered is whether the interpolated (or extrapolated) new Pareto optimal solutions are valid. Since they are estimated (by kriging or the GRNN), the true objective function of solutions of interest must be validated with an exact calculation. The methods presented here could be used with almost any heuristic and with any multi-objective problem for which the objective functions can be calculated. The most important issue here is the quality of the non-dominated solutions acquired from the multi-objective optimization process. Even if there are a few good quality (ie, close to the true Pareto front) non-dominated solutions, using computationally expedient techniques such as kriging or GRNN can be used to enhance the non-dominated solution set as shown in this study.

## References


Agrawal S, Dashora Y, Tiwari MK and Young-Jun S (2008). Interactive particle swarm: A Pareto-adaptive metaheuristic to multiobjective optimization. *IEEE Trans Syst Man Cybern Hum* **38**: 258–277.

Casey K, Lim A and Dozier GV (2006). Evolving general regression neural networks for tsunami detection and response. *Proceedings of the IEEE Congress on Evolutionary Computation* (CEC2006). IEEE Computer Society: Washington, DC, USA, pp 2451–2458.

Coello Coello CA, Lamont GB and Van Veldhuizen DA (2007). *Evolutionary Algorithms for Solving Multi-objective Problems*. 2nd edn. Springer: New York.

Coello Coello CA and Lechuga MS (2002). MOPSO: A proposal for multiple objective particle swarm optimization. *Proceedings of the IEEE Congress on Evolutionary Computation* (CEC2002). IEEE Computer Society: Washington, DC, USA, pp 1051–1056.

Coppersmith D and Winograd S (1990). Matrix multiplication via arithmetic progressions. *J Symb Comput* **9**: 251–280.

Cressie NAC (1993). *Statistics for Spatial Data*. John Wiley & Sons: New York.

Deb K (2001). *Multi Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons: New York.

Fieldsend JE and Singh S (2002). A multi-objective algorithm based upon particle swarm optimisation, an efficient data structure and turbulence. *Proceedings of the 2002 U.K. Workshop on Computational Intelligence*. Birmingham, UK, pp 37–44, http://empslocal.ex.ac.uk/people/staff/jefields/JF_07.pdf.

Garrett A, Dozier GV and Deb K (2007). NEMO: Neural enhancement for multiobjective optimization. *Proceedings of the IEEE Congress on Evolutionary Computation* (CEC2007). IEEE Computer Society: Washington, DC, USA, pp 3108–3113.

Goldberg DE, Korb B and Deb K (1989). Messy genetic algorithms: Motivation, analysis and first results. *Complex Syst* **4**: 415–444.

Kennedy J and Eberhart RC (1995). Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*. IEEE Service Center: Piscataway, NJ, pp 1942–1948.

Kleijnen JPC and van Beers WCM (2005). Robustness of kriging when interpolating in random simulation with heterogeneous variances: Some experiments. *Eur J Opl Res* **165**: 826–834.

Krige DG (1951). A statistical approach to some basic mine valuation problems on the Witwatersrand. *J Chem Metall Min Soc S Af* **52**: 119–139.

Matheron G (1973). The intrinsic random functions and their applications. *Adv Appl Probab* **5**: 439–468.

Miettinen KM (1999). *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers: Boston.

Mitchell TM (1997). *Machine Learning*. McGraw Hill: Boston.

Olea RA (1974). Optimal contour mapping using universal kriging. *J Geophys Res* **79**: 695–702.

Polat O and Yıldırım T (2008). Genetic optimization of GRNN for pattern recognition without feature extraction. *Expert Syst Appl* **34**: 2444–2448.

Reyes-Sierra M and Coello Coello CA (2006). Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *Int J Comput Intell Res* **2**: 287–308.

Sacks J, Welch WJ, Mitchell TJ and Wynn HP (1989). Design and analysis of computer experiments. *Stat Sci* **4**: 409–435.

Schaffer JD (1984). *Some experiments in machine learning using vector evaluated genetic algorithms*. PhD thesis, Vanderbilt University.

Simpson TW, Mauery TM, Korte JJ and Mistree F (2001). Kriging models for global approximation in simulation-based multi-disciplinary design optimization. *AIAA J* **39**: 2233–2241.

Skriver AJV and Andersen KA (2003). The bi-criterion semi-obnoxious location (BSL) problem solved by an ε approximation. *Eur J Opl Res* **146**: 517–528.

Specht DF (1991). A general regression neural network. *IEEE T Neural Networ* **2**: 568–576.

Strassen V (1969). Gaussian elimination is not optimal. *Numer Math* **13**: 354–356.

Trochu F (1993). A contouring program based on dual kriging interpolation. *Eng Comput* **9**: 160–177.

Werner H and Obach M (2001). New neural network types estimating the accuracy of response for ecological modeling. *Ecol Model* **146**: 289–298.

Wu J-D and Ye S-H (2009). Driver identification based on voice signal using continuous wavelet transform and artificial neural network techniques. *Expert Syst Appl* **36**: 1061–1069.

Yapicioglu H, Dozier GV and Smith AE (2006). Neural network enhancement of multi-objective evolutionary search. *Proceedings of the IEEE Congress on Evolutionary Computation* (CEC2006). IEEE Computer Society: Washington, DC, USA, pp 1909–1915.

Yapicioglu H, Smith AE and Dozier GV (2007). Solving the semi-desirable facility location problem using bi-objective particle swarm. *Eur J Opl Res* **177**: 733–749.

## Appendix A

### Test functions

*Test Problem* 1

Objective 1: $\min f_1(\mathbf{X}) = \sum_{i=1}^{n} w_{i1} d(\mathbf{X}, a_i)$

Objective 2: $\min f_2(\mathbf{X}) = \sum_{i=1}^{n} w_{i2} [d(\mathbf{X}, a_i)]^{-1}$

Where $d(\cdot)$ is the Euclidean distance between the fixed point $a_i$ and point $\mathbf{X}$.

*Test Problems* 2 *and* 3

Objective 1: $\min f_1(\mathbf{X}) = \sum_{i=1}^{n} w_{i1} d(\mathbf{X}, a_i)$

Objective 2: $\min f_2(\mathbf{X}) = \sum_{i=1}^{n} e_{i2}(\mathbf{X}, a_i)$

where

$$e_{i2}(\mathbf{X}, a_i) = \begin{cases} M, & \text{if } d(\mathbf{X}, a_i) \leqslant d_1 \\ M - m(w_{i2} d(\mathbf{X}, a_i)), & \text{if } d_1 < d(\mathbf{X}, a_i) < d_2 \\ 0, & \text{if } d_2 \leqslant d(\mathbf{X}, a_i) \end{cases}$$

Where $d(\cdot)$ is the distance (Euclidean for test Problem 2 and rectilinear for test Problem 3) between fixed point $a_i$ and a facility located at point $\mathbf{X}$. $M = 200$, $m = 1$, $d_1 = 10$ *and* $d_2 = 30$.

*Test Problems* 4 *and* 5

Objective 1: $\min f_1(\mathbf{X}) = \sum_{i=1}^{n} w_{i1} d(\mathbf{X}, a_i)$

Objective 2: $\max f_2(\mathbf{X}) = \min_{1 \leqslant i \leqslant n} (w_{i2} d(\mathbf{X}, a_i))$

*Where $d(\cdot)$ is the distance that is Euclidean for test Problem 4 and rectilinear for test Problem 5 between the fixed point $a_i$ and a facility located at point $\mathbf{X}$.*

*Test Problem* 6

Objective 1: $\min f_1(\mathbf{X}) = \sum_{i=1}^{n} w_{i1} [d(\mathbf{X}, a_i)]^{-b}$

Objective 2: $\min f_2(\mathbf{X}) = \sum_{i=1}^{n} w_{i2} d(\mathbf{X}, a_i)$

Where $d(\cdot)$ is the Euclidean distance between the fixed point $a_i$ and a facility located at point $\mathbf{X}$ and $b = -2$.

The data for the first five test problems are given in Table A1.

**Table A1** Data for the first five test problems

| | $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Test Problems 1, 2, 3 | $\mathbf{a}_i$ | (5, 20) | (18, 8) | (22, 16) | (14, 17) | (7, 2) | (5, 15) | (12, 4) |
| | $w_1$ | 5 | 7 | 2 | 3 | 6 | 1 | 5 |
| | $w_2$ | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Test Problem 4 | $\mathbf{a}_i$ | (4, 4) | (8, 7) | (11, 10) | (13, 4) | | | |
| | $w_1$ | 2 | 4 | 3 | 2 | | | |
| | $w_2$ | 1 | 1 | 1 | 1 | | | |
| Test Problem 5 | $\mathbf{a}_i$ | (1, 3) | (4, 5) | (6, 1) | (6, 7) | (8, 5) | | |
| | $w_1$ | 3 | 2 | 3 | 1 | 2 | | |
| | $w_2$ | 1 | 1 | 1 | 1 | 1 | | |

*Note*: Data for Problem 6 is not included in the table above, but can be found in Skriver and Andersen (2003).

# Appendix B

## Pareto set and Pareto front figures

Figures B1–B18.

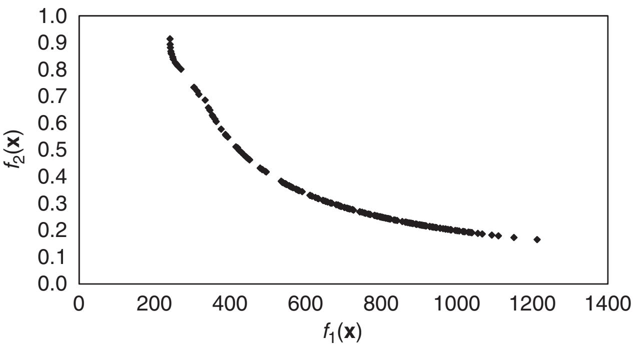


**Figure B1** Pareto front obtained by PSO (Problem 1).

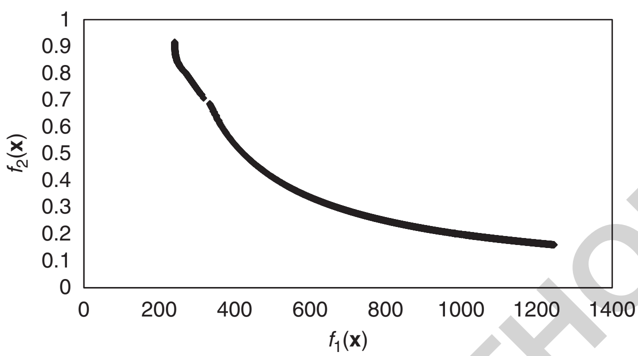


**Figure B2** Merged Pareto front obtained by kriging and PSO (Problem 1).

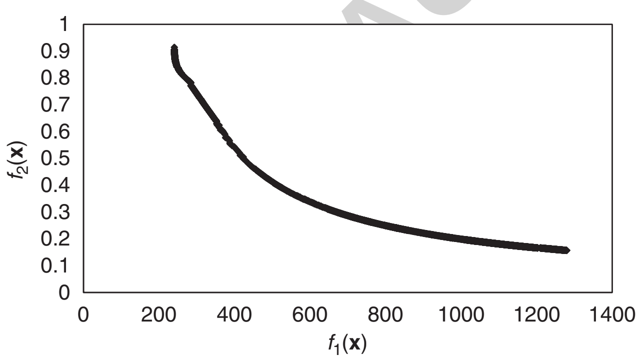


**Figure B3** Merged Pareto front obtained by GRNN and PSO (Problem 1).

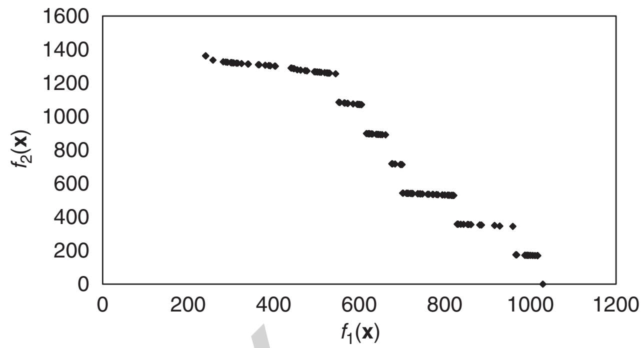


**Figure B4** Pareto front obtained by PSO (Problem 2).

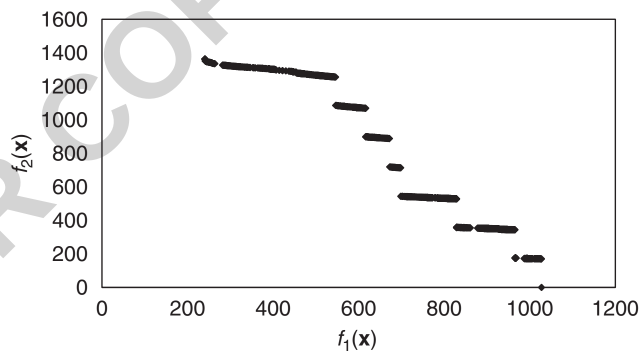


**Figure B5** Merged Pareto front obtained by kriging and PSO (Problem 2).

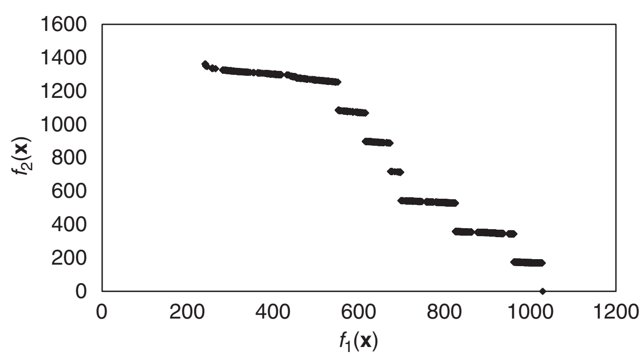


**Figure B6** Merged Pareto front obtained by GRNN and PSO (Problem 2).

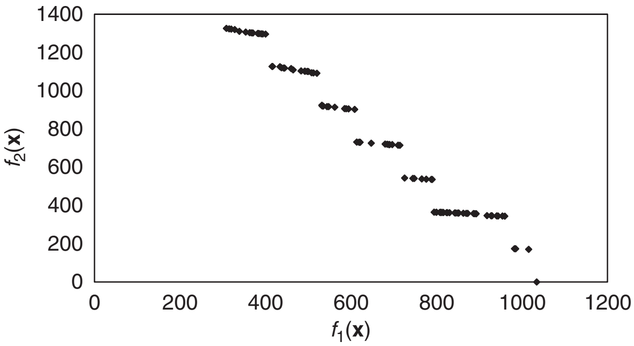


**Figure B7** Pareto front obtained by PSO (Problem 3).

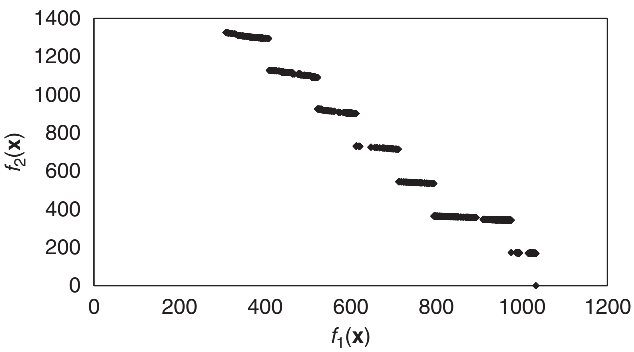


**Figure B8** Merged Pareto front obtained by kriging and PSO (Problem 3).

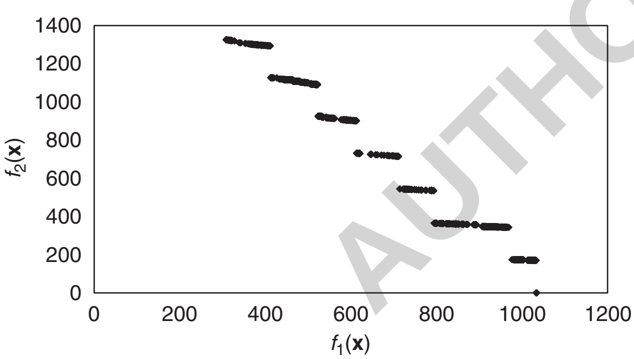


**Figure B9** Merged Pareto front obtained by GRNN and PSO (Problem 3).

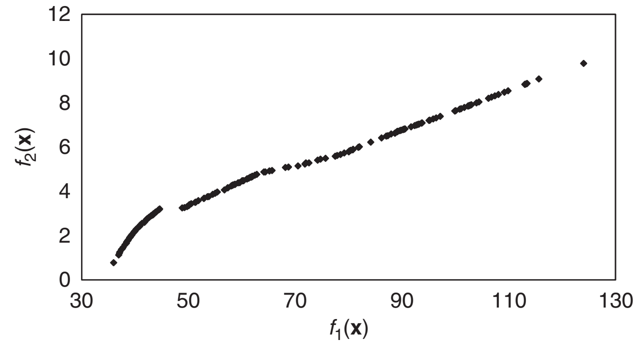


**Figure B10** Pareto front obtained by PSO (Problem 4).

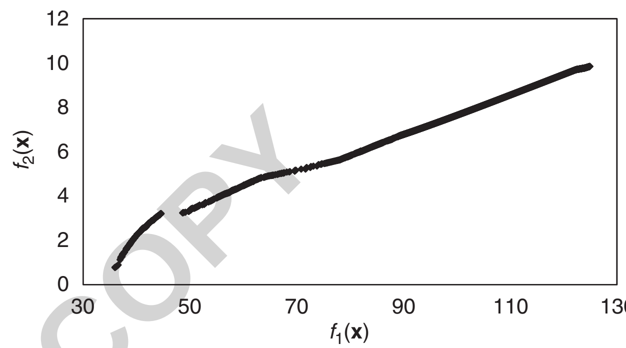


**Figure B11** Merged Pareto front obtained by kriging and PSO (Problem 4).

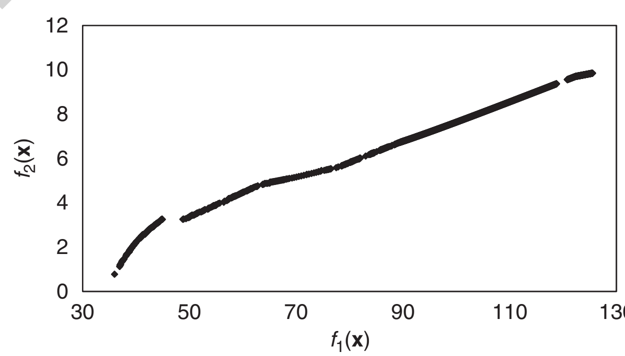


**Figure B12** Merged Pareto front obtained by GRNN and PSO (Problem 4).

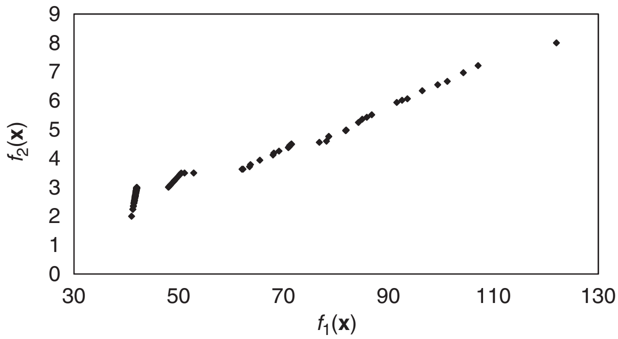


**Figure B13** Pareto front obtained by PSO (Problem 5).

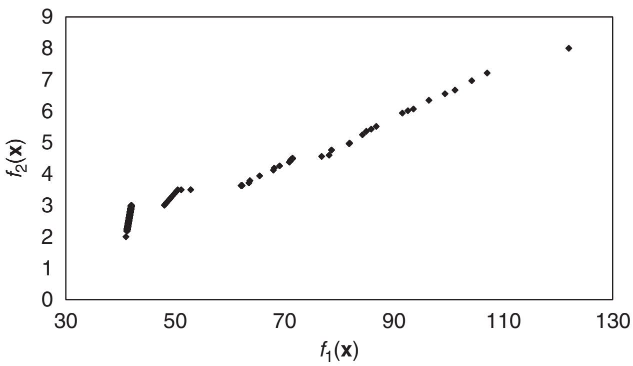


**Figure B14** Merged Pareto front obtained by kriging and PSO (Problem 5).

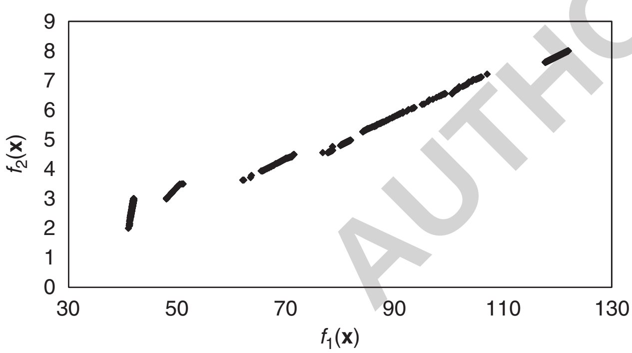


**Figure B15** Merged Pareto front obtained by GRNN and PSO (Problem 5).

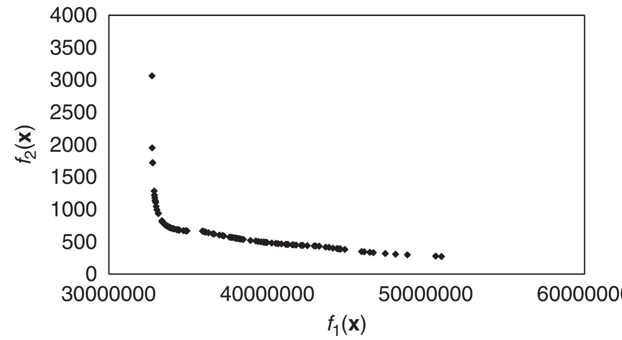


**Figure B16** Pareto front obtained by PSO (Problem 6).

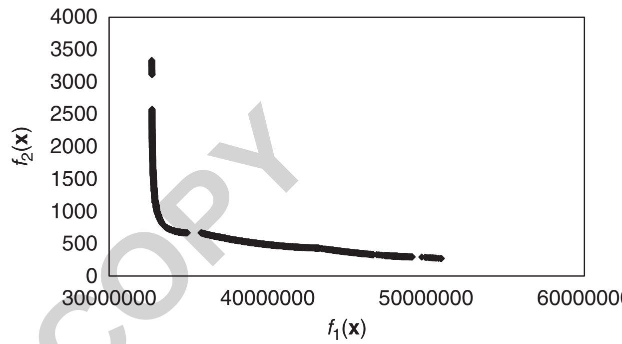


**Figure B17** Merged Pareto front obtained by kriging and PSO (Problem 6).

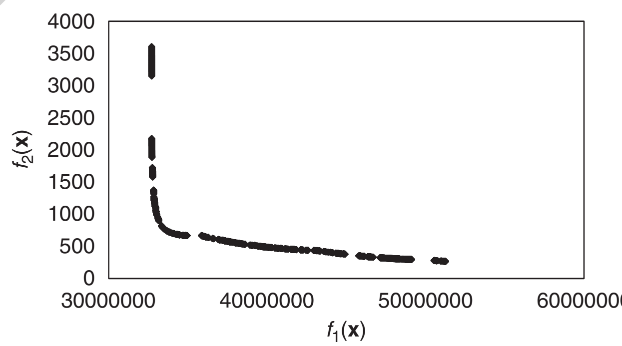


**Figure B18** Merged Pareto front obtained by GRNN and PSO (Problem 6).

*Received June 2009;*
*accepted August 2010 after three revisions*