1 Introduction

The derivative information of a function is immensely valuable for many optimization problems, however to derive a derivative, an optimization problem has to satisfy some mathematical assumptions. In many real-world applications the derivative information is not available or unknown. The design of helicopter rotor is an application in which the approaches that apply derivative information are not practical. This is one of the reasons that why derivative-free optimization methods e.g. meta-heurstic approaches have become popular in the recent decade. One of the advantages of meta-heuristic approaches is that it can be applied to resolve any types of optimization problems. However each meta-heuristic approach may need to be tailored for a particular application to increase speed, reduce the chances for the local minima traps, and return accurate results. As an example, in a real time optimization, speed has a higher priority compared to the accuracy of result or trap of local minima. On the other hand, in off-line optimization, the speed might not be a serious factor as long as an optimum solution is obtained in a feasible time. In this context, we propose and investigate different hybridization frameworks with their specific type of optimization problem for which they are designed. As an example the design of a pressure vessel with minimum cost of production is studied with the proposed hybridization framework.

Fuzzy linear regression analysis is a powerful tool for data analysis with uncertainties in which the collected data are inexact. As an example one could refer to classification of quality of a service, product, resource and so on. In conventional approaches the quality may be quantized, e.g. by an expression such as 3-star hotel, and premium meat. Quality values are good indicators for average costumers or clients, however, they do not provide more information that may exist in the meaning of these values. A model derived from a fuzzy regression analysis task can take the uncertainty into consideration. As in any other optimization problem, the selection of the most suitable objective function for a particular application in fuzzy linear regression analysis is a very important factor. However, majority of the approaches proposed in the literature for fuzzy linear regression impose some limitations on the type of the objective function. This is where the application of meta-heuristic approaches is useful. Other related topics such as detecting outliers and tackling the spread increasing problem can be solved by using meta-heuristic approaches. Application of fuzzy regression is given for predicting the quality of service for web services, where it is shown how fuzzy regression can be applied for quantized data and still take the uncertainty into consideration.

1.1 Overview

In many real-world applications, a function which relates the input to ouput variables of a given system is required to be defined. This function is then applied for forecasting or is used for further study of the behaviour of the system. Linear functions are perhaps the simplest forms of functions for describing the relationship between the input variables and output variable where the linearity assumption exists. In the general case where the data comes with uncertainty originating from experts' opinions, a Fuzzy Linear Function (FLF) is a suitable technique for representing the relationship of the input and output variables. Fuzzy Linear Regression Analysis (FLRA) is a powerful tool to derive the fuzzy linear function for variables with uncertain nature.

FLRAs ie especially useful for data sets which are generated based on the subjectivity of human decisions [5] or abstracted due to simplification purposes. The application of FLRA to estimate the relationship of a given data and then to predict a continuous response to new data is reported for many engineering problems [6].

To derive the best FLF which describes the relationship between input and output variables, a Model Fitting Measurement (MFM) has to be defined. MFM is an essential tool to measure the goodness of a defined model where the similarity between each of predicted and estimated outputs is computed. FLRA aims at seeking for a FLF which maximizes the model fitting measurement value.

Least Squares (LS) and Linear Programming (LP) are the two main approaches of FLRA. In both LS and LP approaches, the MFM has to be defined in advance and must be transparent to designers. The transparency is required otherwise designers cannot consider the conditions that have to be taken into consideration for modeling LS or LP approaches. However, the transparency of the MFM is dependent on the confidentiallity level of the MFM. The MFM could be a trade secret where revealing its formulation could be risky for the entity who owns it. The LS and LP based approaches are inapplicable in such situations as in both approaches a clear formulation of the MFM is required. The only way in which the MFM owner does not need to reveal the formulation is when the input-output behaviour of the MFM is provided. So as an alternative, designers could solve FLRA by applying approaches that do not need the mathematical formulation of the MFM.

In real-world applications where the global optimum of a function is sought, nonlinear programming techniques can be applied. The application of nonlinear programming techniques may fail if an optimization problem has multiple local minima [16]. Moreover the gradient descent based approaches are not always applicable due to non-differentiability of the function or unknown formulation of the function [17]. For such optimization problems where classical optimization techniques are not applicable, meta-heuristic approaches which mimic the nature or intelligent procedures are suitable tools [16, 17]. Fuzzy linear regression fits some of the characteristics mentioned above, i.e., when a model fitting measure is a trade secret [6], outliers exists in the data set [8], and the spreads of the model increase with increase in the magnitute of input data [9]. The objectives of this research are identified as follows:

- To investigate properties of optimization problems which make them suitable to be solved by global continuous optimization based on meta-heuristics.
- To investigate the situations where the application of global continuous optimization approaches is a necessity for fuzzy linear regression.

1.2 Contributions

In this thesis, we define the properties of optimization problems which make them suitable to be solved by global continuous optimization. Some examples of these properties include *unreliable derivatives* and/or *black-box nature*. We then define the draw-backs of global continuous optimization methods based on meta-heuristic approaches. As a solution we propose hybridization which may improve one or more of the success rate, average error, and number of function evaluations. For comparison of the global continuous optimization we introduce a comparison protocol. All of the optimization problems discussed in this thesis are limited to single objective optimization. The application of multi-objective optimization is discussed as part of future work.

The application of global continuous optimization is then applied for fuzzy linear regression. Three challenges in fuzzy linear regression of non-transparent fitness measure, outlier detection or removal, and spread increasing problem, are identified to be tackled by global continuous optimization. Fuzzy polynomial regression and non-parametric regression are discussed as part of future work. For comparing the models we introduce a comparison protocol which compares two models for a given data set with regards to a finite number of fitness measures. We compare and verify the performance our proposed method on two main categories of data; fuzzy input-fuzzy output, and crisp input-fuzzy output. We also define a comparison protocol to compare the superiority of the models with regards to different fitness measures. Both outlier detection or removal and spread increasing problem are tackled in the scope of fuzzy linear regression, however, a similar concept may be applied for fuzzy polynomial and non-parametric regression which is not in the scope of this research. The proposed outlier removal approach is verified and compared with existing methods on small and large crisp input-fuzzy output data sets which are taken from the literature. The small data set has three variations of constant, decreasing and increasing spreads and the large data set has random spreads. Spread increasing problem is discussed and then an approach based on global continuous optimization is proposed to tackle the problem. The approach is also compared to models proposed to solve spread increasing problem in the literature with regards to number of parameters, and flexibility of the spreads.

1.3 Organization

The results of this PhD study have been published in several international conferences and two archival journals.

Majority of the materials in Chapter 2 are taken from the following paper:

• M. H. Mashinchi, M. A. Orgun, and W. Pedrycz. *Hybrid optimization with improved tabu search*. Applied Soft Computing 11(2), 1993-2006 (2011).

In Chapter 2 we discuss the application of global continuous optimization for optimization problems with objective functions with one or more of following characteristics: unreliable derivatives, non-transparent nature, computationally expensive evaluations, costly (financial-wise) or dangerous evolutions, and/or optimizations with time constraints. Different approaches of global continuous optimization are then briefly studied. As a common drawback in all of the GCOMs, meta-heuristic approaches do not guarantee finding global optimum solutions, but they can be used for a wide range of optimization categories without considering strict assumptions. Moreover most of the time they can avoid the traps of local minima. The main drawback of some of the meta-heuristic approaches for GCOPs is their slow rate of convergence if a very detailed solution is required. In other words, they can find a promising valley (a local minimum area) but getting to the bottom of the valley is a time consuming process. To overcome some of the above mentioned issues, we investigate different hybridization approaches. As an example we apply one of the hybrid approaches for a set of benchmark functions and also an industrial engineering application and then we compare the results with other methods in the literature (see Appendix A.1). Finally the application of global continuous optimization is motivated for fuzzy linear regression analysis.

Majority of the materials in Chapter 3 are taken from the following papers:

• M. H. Mashinchi, M. A. Orgun, M. Mashinchi, and W. Pedrycz. *A tabu-harmony search based approach to fuzzy linear regression*. IEEE Transactions on Fuzzy Systems 19(3), 432-448 (2011).

• M. H. Mashinchi, M. A. Orgun, and M. Mashinchi. *Solving fuzzy linear regression with hybrid optimization*. In 16th International Conference Neural Information Processing, 336-343 (2009).

In Chapter 3 we introduce the mathematical preliminaries for fuzzy set theory used throughout this thesis. The mathematical formulation and different approaches in fuzzy linear regression analysis are then studied. The application of fuzzy regression is proposed for problems with one or more of following characteristics: abstraction has occurred due to high complexity of environment, human subjectivity is involved in the system, or the information is partially available. As an important part of fuzzy linear regression, some of the similarity measures in the literature of fuzzy linear regression are investigated. Linear programming and least square based approaches with their advantages and disadvantages are dicussed. Finally the application of global continuous optimization is motivated for three main issues in fuzzy linear regression as follows; application of fitness measure, outlier detection and spread increasing problem.

The majority of the materials in Chapter 4 are taken from the following papers:

• M. H. Mashinchi, M. A. Orgun, M. Mashinchi, and W. Pedrycz. *A tabu-harmony search based approach to fuzzy linear regression*. IEEE Transactions on Fuzzy Systems 19(3), 432-448 (2011).

• M. H. Mashinchi, M. A. Orgun, and M. Mashinchi. *Solving fuzzy linear regression with hybrid optimization*. In 16th International Conference Neural Information Processing, 336-343 (2009).

In Chapter 4 we introduce a hybrid global continuous approach based on metaheuristic approaches for fuzzy linear regression analysis. We propose a protocol for comparing fuzzy linear regression analysis methods for a given data set. Based on this protocol, a model is not necessarily better or worse than another method, yet it could be Pareto-equivalent to another model based on chosen fitting measures for a given data set.

The majority of the materials in Chapter 5 are taken from the following paper:

• M. H. Mashinchi, M. A. Orgun, M. Mashinchi, and W. Pedrycz. *A tabu-harmony search based approach to fuzzy linear regression*. IEEE Transactions on Fuzzy Systems 19(3), 432-448 (2011).

In Chapter 5 we investigate the performance of our proposed approach compared to other fuzzy linear regression approaches reported in the literature. For comparison purposes, we report the performance of the model for four different fitting measures available in the literature. The verification of the model is done by testing the model on two different categories of fuzzy data sets as; fuzzy input-fuzzy output, crisp input-fuzzy output.

The majority of the materials in Chapter 6 are taken from the following paper and a techinical report:

• M. H. Mashinchi, M. A. Orgun, and M. R. Mashinchi. *A least square approach for the detection and removal of outliers for fuzzy linear regression*. In World Congress on Nature & Biologically Inspired Computing, 134-139 (2010).

• M. H. Mashinchi. *Application of hybird optimization for spread increasing problem*. Technical Report, Macquarie University, 2011.

In Chapter 6 we tackle outlier detection and spread increasing problem in fuzzy linear regression analysis. In the first part, we investigate the proposed approaches in the literature with regards to outlier detection or removal and then compare and contrast the methods. Then we propose an approach which aims at solving the issues that currently exist in outlier detection or removal approaches in the literature such as having multiple user-defined variables and being computationally very expensive. The second part of this chapter provides some definitions for the spread increasing problem. Then an approach based on the application of global continuous optimization is proposed for fuzzy linear regression analysis which aims at solving the spread increasing problem.

The majority of the materials in Chapter 7 are taken from the following papers:

• M. H. Mashinchi, L. Li, M. A. Orgun, and Y. Wang. *The prediction of trust rating based on the quality of services using fuzzy linear regression*. In IEEE International Conference on Fuzzy Systems, 1953-1959 (2011).

• M. H. Mashinchi, M. A. Orgun, and W. Pedrycz. *Hybrid optimization with improved tabu search*. Applied Soft Computing 11(2), 1993-2006 (2011).

In Chapter 7 the application of global continuous optimization as well as fuzzy linear regression is studied for real-world problems. The cost minimization of producing a pressure vessel is given as an optimization problem. Then the fuzzy linear regression analysis which uses global continuous optimization is applied for quality of service prediction in web services. The results are compared with those of the other methods available in the literature.

Finally Chapter 8 discusses the conclusions and future work. Some of the promising directions for future work are gathering real fuzzy data, an application of global continuous optimization for non-parametric regression, and an application of multi-objective optimization for fuzzy linear regression with more than one fitting measure to be optimizated.

2
Global continuous optimization[1]

The application of global continuous optimization methods is a necessity in many engineering applications where an optimization problem has certain properties such as *unreliable derivatives* and/or *black-box nature*. *Meta-heuristics based approaches*, *direct search strategies* and *surrogate models* are the main approaches proposed to tackle such optimization problemes. Meta-heuristic based optimizations, as one of the promising approaches in global continuous optimization, have a slow rate of convergence. Hybridization frameworks are then investigated as a potential way of enhancing the optimization speed, and the quality of solutions.

2.1 Global continuous optimization

Finding the global minimum of an unknown system (function or objective function), which may have many unknown local minima is one of the significant problems that arise in many engineering applications. The only information which is obtainable in these unknown systems is the input-output behaviour. As examples in engineering applications; one can point at fluid power circuits [18], electronic circuit designs [19], balancing of hydraulic turbines [20], computational chemistry [21],

[1] This chapter is based on a journal paper; [2] co-authored by M. A. Orgun and W. Pedrycz.

developing optical fibers [22], signal setting problems [23], non-destructive control [24], optimization of electromagnetic devices [25] and others. The diversity of applications has led to a great deal of interest to develop Global Continuous Optimization Methods (GCOM) which are capable of finding the global minimum solution in a continuous search space.

A GCOM searches for an input vector, x, in the search space such that the system's output (reaction) reaches its minimum. In such problems a local minimum is not acceptable and the GCOM should be able to avoid being trapped in local minima and return a global minimum solution instead. The formulation of the unconstrained global minimization problem is as follows:

$$\begin{aligned} &y = \min\ f(x) \\ &f : \mathbb{R}^n \rightarrow \mathbb{R}, \end{aligned}$$

where $f(x)$ can be either a convex or non-convex real-valued unknown system [26, 27]. Note that there is no need to investigate a global maximization problem separately, since it can be converted to a global minimization problem in a standard manner [27]:

$$\begin{aligned} &\max\ f(x) = -\min[-f(x)] \\ &f : \mathbb{R}^n \rightarrow \mathbb{R}. \end{aligned}$$

Note that the value obtained from minimization process on $-f(x)$ is the reflection (with respect to x-axis) of maximum point on the original function $f(x)$, and so its sign has to be changed. The change of sign is done via the minus sign in front of the min function.

Due to the lack of information about the behaviour of an unknown system, no specific assumptions are being made. Thus it is difficult to solve them using analytical methods such as complete enumerative search strategies, homotopy methods, branch and bound algorithms or gradient-best approaches, all of which require certain conditions and can be only applied to a specific group of problems [28]. There are no efficient approaches to solve these kinds of problems in general, especially those of high dimensionality [29]. As an example, note that derivative-based approaches can only be applied for differentiable systems and there is no guarantee to reach the global minimum. Although analytical approaches are still being investigated for potential applications in the Global Continuous Optimization (GCO), but their applications are fading, because in reality little (or sometimes no) detailed information is available about the optimization problems.

2.2 When to use global continuous optimization

Due to a variety of reasons such as unavailability and/or unreliability of the derivatives of many optimization problems, standard mathematical approaches are not applicable [30, 31]. Moreover in some optimization problems the derivative information is *practically not feasible*. By this, we mean it is either not computable as no assumptions can be made regarding the differentiability or continuity of the function or it is very computationally expensive to estimate the derivatives. In some other type of optimization problems, the cost of computing the objective function values in terms of both money and computation time is very expensive. *Derivative-free optimization* methods are nonlinear optimization techniques which are suitable for the class of problems when derivative information is unavailable, unreliable, impractical to obtain, or it is noisy [30, 31]. Such optimization problems can be found in many applications such as engineering design [32–34], chemistry [35] and biology [36].

We discuss the design of a helicopter rotor blade while minimizing the vibration transmitted to the hub. This problem comes from the Helicopter Division of The Boeing Company [30, 32, 37]. The simulation is multidisciplinary and involves, aerodynamics, fluid dynamics, structures and propulsion disciplines. This is a non-convex optimization problem which has many local minima and is not defined everywhere and may not be smooth everywhere it is defined.

One main category of optimization problems contains problems which are characterized by one or more of the following properties of

Unreliability of the derivatives: Due to the existence of noise in the objective function or other reasons, the derivatives may be highly unreliable and/or completely useless [16, 30, 38]. As an example, consider an engineering system which generates an output based on some input values, however the output is not necessarily precise and may come with a random noise. The noise could come from the nature of the system itself or a measurement or recording error. In the minimization of the vibration for the helicopter rotor blade design, if a high fidelity level of the objective function is required, then a single evaluation could take up to days [30, 32]. However, if a quick evaluation is demanded, then the fidelity of the objective value is low which suggests that it comes with high noise. Consequently the gradients become unreliable as they are estimated according to the objective values which contain noise. Other examples can be found in optimization of molecular geometries [39, 40], and shared computing networks [41].

Non-transparent nature: Due to the black-box nature of the objective function, no assumptions can be made with regards with differentiability and the continuity of the objective function. The black-box nature may come from codes which have been written in the past and have not been maintained by the original authors. Although one could try to re-write or manipulate such codes to provide first-order derivatives, this could be an extremely time-consuming task [30]. Moreover, legacy or proprietary codes necessitate the code to be treated as a non-transparent objective function, for example, in the case when a company who owns the code provides only the binary or object files [30,38]. In the minimization of the vibration in helicopter rotor design, the objective function is provided as an object code which belongs to the Boeing Company [32].

It also worth mentioning that *derivative-free optimization* is often referred to as *optimization over black-boxes* in the literature [31].

Optimization problems with the above properties can be solved by brute force or exhaustive search strategies. But, if the problem has extra properties in addition to the one given above, then the application of brute force or exhaustive search strategies is not practical. The application of derivative-free optimization (which in this thesis is also referred to as global continuous optimization) is preferred for problems which in addition to the above properties, have one or more of the following properties [31].

Computationally expensive evaluations: Computing the objective function or the derivatives is theoretically possible but practically impossible as it is computationally very expensive [16, 38]. In engineering design, there are two main reasons why the function evaluation is very computationally expensive; the engineered function is very complex, and/or high accuracy (fidelity) of the function evaluations is required [32]. In the helicopter rotor design example, a single evaluation of the objective function depending on the fidelity level may take from several minutes to days [30, 32]. It is clear that when function evaluations are computationally expensive, then the finite-difference derivative approximation is inappropriate. This is due to the fact that for a single gradient estimation normally no less than the number of variables plus one evaluation are required [30].

Costly (financial-wise) or dangerous evaluations: Evaluating an objective function could be financially very expensive. As an example, function evaluations can be very financially costly in physical processes where a group of people have to conduct some simulation with the defined quantity of some materials to be able to evaluate the result. Considering that the cost of human resources and the materials used in the simulation is high, each function evaluation can cost a lot of money. Also, the simulation could be a dangerous process (maybe for some input values). Due to the costly and/or dangerous nature of function evaluation in such optimization problems, obtaining a global optimum is not necessarily a *must* and a better solution to the current system or an acceptable solution can be satisfactory [38].

Optimization with time constraints: In many engineering and science applications, obtaining a solution in extremely long time is a negative factor, especially when a shorter design cycle time, and a faster turn-around can improve the result of the whole engineering process [38]. In some real-time applications, a time constraint may dictate to obtain an acceptable solution before a specific deadline. However, obtaining a solution before the deadline (faster) could be even more advantageous as it leaves more time for other dependant procedures to be done. Although function evaluations can be computationally inexpensive, and/or not costly or dangerous, the optimization procedure should be able to find an acceptable solution within a certain period of time.

2.3 Global continuous optimization approaches

Global continuous optimization methods can be classified into *direct* and *indirect* approaches. In direct approaches, the current function evaluation/s is used directly for the next solution whereas indirect approaches utilize a surrogate model of the objective function based on the current function evaluations [31]. Direct approaches can be classified as stochastic and deterministic, depending on whether they take random steps from the current state to the next states or not [16,31] . In the following sections, we refer to *stochastic direct*, *deterministic direct*, and *direct approaches* as *meta-heuristic based approaches*, *direct search strategies*, and *surrogate models*, respectively. Table 2.1 shows the innovation time line of the main global continuous optimization methods which are discussed in this thesis.

2.3.1 Meta-heuristics based approaches

Meta-heuristic approaches heuristically generate the next solution/s in a non- deterministic algorithmic manner [31]. The generation of the next solution/s is designed to be able to explore the search space as extensively as possible. Depending on how the next solution/s are generated, meta-heuristic approaches are classified into *population-based*, or *point to point* approaches [2,16].

Table 2.1: Timeline of innovation in global continuous optimization

Method	Year of pulication	Note
Hooke-Jeeves algorithm [42]	1961	Journal paper
Nelder-Mead algorithm [43]	1965	Journal paper
Genetic algorithms [44]	1975	Text-book
Simulated annealing [45]	1982	Journal paper
Tabu search [46]	1989	Journal paper
Derivative-free optimization [47] (surrogate models)	1997	PhD thesis
Derivative-free optimization [32] (surrogate models)	1999	PhD thesis
Harmony search [48]	2001	Journal paper
Global continuous optimization [16] (meta-heuristics)	2004	PhD thesis
Global continuous optimization [30] (surrogate models)	2009	First DFO text-book
Global continuous optimization [31] (surrogate models)	2009	PhD thesis Application to protein-ligand docking problem
Global continuous optimization [38] (surrogate models)	2009	PhD thesis

Due to the ill-defined nature of optimization problems, and the weakness of mathematical approaches, there is a growing interest in meta-heuristic search [29]. Also, optimization problems do not need to satisfy strict requirements of differentiability. It is advantageous as in the real world applications, we usually end up with optimization of an non-differentiable system as an example signal setting problem can be addressed [23].

As a common drawback in all of the GCOMs, meta-heuristic approaches do not guarantee finding global optimum solutions, but they can be used for a wide range of optimization categories without considering strict assumptions. Also most of the time they can avoid the traps of local minima [49]. Among all meta-heuristics, Tabu Search (TS) stresses emphasis on escaping from local minima by introducing the concept of a list. In addition to that list, TS allows moving to another solution from the current solution even though it allows for a higher value of the objective function in the anticipation that it will achieve a better solution in the consecutive iterations [50].

The common drawback of some of the meta-heuristic approaches for GCOPs is their slow rate of convergence if a very detailed solution is required. In other words, they can find a promising "valley" (local minimum area) but getting to the bottom of the valley is a time consuming process. To overcome this issue, meta-heuristic approaches are mostly hybridized with Local Optimizer Methods (LOM). At the beginning a meta-heuristic approach is applied to escape from local optima and determine the promising areas. This process is usually known as diversification (exploration). Then a LOM starts from such promising areas obtained from the result of running diversification. It searches for a global optimum solution around promising areas. This process is known as intensification (exploitation).

In the following sections two population-based meta-heuristics; genetic algorithms and harmony search and two point to point meta-heuristics; tabu search and simulated annealing are described.

2.3.1.1 Genetic Algorithms

Genetic algorithms (GAs) are inspired from the genetic evolution of a species, proposed by Holland in 1975 [44]. They start from a random population whose elements are chromosomes and in each generation the chromosomes are substituted with new chromosomes based on three operators of selection, crossover, and mutation. The selection operator choses the chromosomes for further manipulation based on their performance which is computed by the objective function. The mating of the chosen chromosomes is done by the crossover operator and aims at generating better chromosomes. Also similar to the evolution of the species in the nature, a small portion of the generated chromosomes are mutated. GAs are stochastic methods as the selection operator works based on the probability of individuals' fitness, and also the mutation is done randomly. With optimization by GAs, the search space can be extensively explored as in each generation a pool of chromosomes is generated. However, as almost all the chromosomes change in each generation, the number of function evaluations are usually reported to be high [2]. This can be a negative factor when the objective function has the propertie of *computationally expensive evaluations*, *costly or dangerous evaluations*, or *optimization with time constraints* as given in Section 2.2.

GAs are one of the most applied meta-heuristics for black-box optimization problems [16]. The objective function does not need to satisfy any condition such as differentiability or continuity. In the general case, the underlying function to be minimized can be *non-transparent* (as described in Section 2.2) which makes it suitable for derivative-free optimization problems [16, 51]. However, it has been observed that GAs are could perform better for intensification if they are hybridized with local optimizer methods–a topic which will be discussed in Section 2.4.

2.3.1.2 Tabu search

Glover [46, 52] proposed the idea of TS for combinatorial optimization problems. The original method was then adapted to continuous optimization problems by Hu [53], known as CTS. TS is preferred for optimization problem with low dimension as it is a point to point search strategy. However, it still can explore the search space extensively as it can keep track of the previous explored areas.

TS is an iterative optimization method which starts from an initial random solution (point) in the pre-defined search space. Then it generates random neighbours around the current point and moves from the current solution to the best neighbour if we have not seen it recently. To keep track of recent points, TS has a limited memory called Short-Term Tabu List (STTL). The list works in first-in first-out manner and avoids TS to cycle around local minima [54]. There is another limited memory that keeps track of the best seen points so far. This is the Long-Term Tabu List (LTTL), also known as "promising list" in the literature. LTTL is implemented for diversification and expands the search area [54]. So, once a good solution is detected, it is stored in LTTL which means that the surrounding area of this point has already been explored. The process of generating neighbours iterates until a certain stopping criterion has been met. In contrast with hill-climbing approaches, in TS the next point can have worse performance than the current one. This property also helps TS to avoid trapping in the local minima.

There are two major differences between CTS and the basic TS. First, the mechanism for generating neighbours around the current solution in TS is easier than the one used in CTS. Since the search space in the basic TS is discrete, only a finite number of neighbours exists around the current solution. Thus, all the neighbours can be considered for the next move. But in CTS, each current solution has an infinite number of neighbours which makes the investigation of the neiborhood area very difficult. The strategy of generating neighbours around the current solution in CTS is known as the neighbour-search strategy. The most basic neighbour-search strategy is to generate a number of random neighbours within a radius of "Distribution Factor" (DF) [55]. In this strategy a finite number of neigbours are generated randomly in a ball with the center of the current point and with radius of DF. This radius plays an important role on the search performance. A very large value of DF causes CTS to act like a pure random search method while a very small DF leads to a local optimization. Thus it is important to find a proper DF. The second difference between CTS and original TS is in the way of tabulating visited solutions. In the TS, if a solution is met then it can be simply tabu for a period of time, but in CTS since the search space is continuous, the algorithm should tabu any solution in a neighborhood space around the visited solutions. This neighbourhood space is a ball with radius of "Similarity Factor" (SF) with the visited solution in the center of the ball [55]. If SF is set to a very large value then the CTS cannot find a near optimum solution. On the other hand, a very small SF makes CTS to search locally rather than globally. Thus selecting the value of SF is as just crucial as DF.

In the basic TS the first step is to initialize the parameters. To start from a point in the search space R random neighbours are generated and the best one is chosen as the starting point and is copied to current-solution (C) and best-solution (S^*). The STTL is updated with C. Then N neighbours are generated randomly based on neighbour-search strategy around C and ranked according to their performance. Then the best neighbour is selected and copied to C if it has not been seen before. This condition can be checked by searching the searched solutions which are stored in STTL. A tabu neighbour can be accepted as a next move if it outperforms the S^*. This is known as aspiration condition. Also for future intensification, S^* should be stored in LTTL. So if C has better performance to S^*, S^* is replaced by C. The process of generating random neighbours and selecting the best ones is continued until a specific number of iterations or if the algorithm does not find a better solution than S^* after a certain number of iterations.

2.3.1.3 Harmony search

Harmony search is a population-based meta-heuristic approach, which adopts the idea of natural musical processes [56]. The algorithm of harmony search minimizes an objective function of the form $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Harmony search is a good search strategy for optimization problems with low dimension or in intensification where a near optimal solution is already sought.

In the basic harmony search, randomly generated available solutions are initialized in the Harmony Memory (HM). In each iteration, the algorithm aims at improvising the HM. The improvision process works based on three operations; memory consideration, pitch adjustment and random selection. In the memory consideration and the random selection operations, each variable of a new solution vector is generated either based on the solutions in HM or randomly. The pitch adjustment operation, introduced to escape from local minima, makes random changes to some of the generated solutions [56, 57].

The algorithm of the basic harmony search can be outlined as follows [56]:

1. Initialization of control variables.

2. One way of initializing the Harmony Memory (HM) with HMS (Harmony Memory Size) is to randomly generate a start-up solution. The start-up solution like $X_t = (x_t^1, x_t^2, \cdots, x_t^{n-1}, x_t^n)$ plus $HMS - 1$ solutions which are chosen randomly in the neighborhood of X_t, generate the HM as given in Algorithm 2.1 (taken from [2]).

Algorithm 2.1 Initializing the harmony memory (taken from [2])

```
for $i = 1$ to $HMS - 1$ do
    for $j = 1$ to $n$ do
        $x_j^i \Leftarrow x_t^i + RND(a,b)$;
    end for
end for
```

Where a and b are real numbers defined based on the domain of the input variables or etc.

Then the solutions are sorted according to the output which is produced by the objective function, as shown below:

$$HM = \begin{bmatrix} x_1^1 & \dots & x_j^1 & \dots & x_n^1 \\ x_1^2 & \dots & x_j^2 & \dots & x_n^2 \\ : & : & : & : & : \\ x_1^i & \dots & x_j^i & \dots & x_n^i \\ : & : & : & : & : \\ x_1^{HMS-1} & \dots & x_j^{HMS-1} & \dots & x_n^{HMS-1} \\ x_1^{HMS} & \dots & x_j^{HMS} & \dots & x_n^{HMS} \end{bmatrix}$$

where $X_j^i = (x_1^i, x_2^i, \cdots, x_j^i, \cdots, x_{(n-1)}^i, x_n^i)$ and n is the number of variables to be optimized.

3. A new HM is improvised from the current HM. So, for each solution vector X^i, $\acute{X}^i$ represents the new solution vector which is going to be substituted for X^i. The next solution can be either chosen from the HM with the probability of the harmony memory consideration rate ($HMCR$), or generated rondomly with the probability of $1 - HMCR$ in the feasible search space M. some proposed values for $HMCR$ is set to be 0.9 as it is an intensification phase and we do not need too much mutation [2]. This solution is then adjusted with a probability of pitch adjustment rate PAR and with the probability of $1 - PAR$ the solution remains intact. In pitch adjustment, the solution changes slightly in the neighborhood space of the solution. To be able to search the neighborhood of each solution comprehensively, we assign a large value like 0.8 to PAR. The entire process of improvising harmony memory can be summarized as follows:

Harmony memory consideration:

$$\acute{X}^i = \begin{cases} X^i \epsilon HM & \text{with probability of } HMCR \\ X^i \epsilon M & \text{with probability of } (1 - HMCR) \end{cases}$$

Pitch adjustment:

$$\acute{X}^i = \begin{cases} \acute{X}^i = \acute{X}^i + (bw \cdot RND(-1,1)) & \text{with probability of } PAR \\ X^i & \text{with probability of } (1 - PAR) \end{cases}$$

where bw is an arbitrary positive value for pitch adjustment which is usually assigned a value less than 1. In some works, bw is set to 0.1. The function $RND(-1,1)$ generates a vector of random numbers coming from the uniform distribution over $[-1,1]$ [2]. The role of $bw \cdot RND(-1,1)$ is to produce both negative and positive values randomly. So if we assign an arbitrary value in the closed interval of $[-1,1]$ to bw instead of $[0,1]$, then we still need to produce random values in the interval of $[-1,1]$.

4. The new solution vector $\acute{X}^i$ is substituted with the worst solution in the HM – provided it outperforms the worst one.

5. If the stopping criterion is not met then GOTO 3.

2.3.1.4 Simulated annealing

Initially proposed to handle combinatorial optimization problems [45], the simulated annealing algorithm was later extended to continuous problems. The algorithm successively generates a solution in a neighborhood of the current solution and based on a probability depending on the difference between their function values determines whether or not the current solution is replaced by the trial point. As a result, simulated annealing allows moves to points with worse objective function values than the current one. By the cooling schedule, simulated annealing is capable of doing both diversification and intensification which will be discussed later in this chapter. The main control parameter in the cooling schedule is the temperature parameter where it is set to be high for diversification and becomes lower (close to zero) in the final stages for the purpose of intensification [16]. The algorithm of the basic simulated annealing optimization method is given in Algorithm 2.2 (taken from [2]).

Simulated annealing does not guarantee finding an acceptable solution in a finite number of iterations [58]. Finite-time performance is of particular interest for optimization problems with *costly or dangerous evaluations*, or *optimization with time constraints* as discussed in Section 2.2 [59].

Algorithm 2.2 Basic simulated annealing algorithm (derived from [16])

```
Initialize tempreture ($T$), Starting point $x_0$
$k = 0$
while Stopping criterion is not met do
    Generate a trial neighbor $y_k$ around the current point $x_k$
    if the function evaluation value at $y_k$ is better than the function evaluation at point $x_k$ then
        Move to the new point $y_k$
    else
        with the probability of $1 - \exp\frac{f(x_k)-f(y_k)}{T}$, $x_{k+1} = y_k$, otherwise stay in the same point $x_{k+1} = x_k$
    end if
    $k = k + 1$
end while
```

2.3.2 Direct search strategies

Direct search methods seek the optimum solution via a sequential examination of trial solutions through the geometric intuition of function evaluation without using gradients [16, 31, 42]. Due to the simplicity and flexibility of direct search methods even after about half a century after the Hooke and Jeeves method was first introduced [42], these methods are still popular. In the following sections, Nelder-Mead [43] and Hooke-Jeeves [42] search strategies are described.

2.3.2.1 Nelder-Mead

Simplex Nelder-Mead strategy is used to find the local minimum point of a function. Considering a function of n variables, initially n points $(P_1, \ldots, P_n)$ in n-dimensional space are generated randomly around a pre-defined point P_0. The pre-defined point P_0 can be chosen either randomly or obtained as a result of another algorithm which is run before. These $n+1$ points are called current simplex, S_0. The first n points define the vector direction that span the n-dimensional space around P_0 [43].

The value of the function in P_i is denoted by Y_i. The highest and lowest values of Y_i are denoted by Y_h and Y_l, respectively and are computed as follows:

$$\begin{aligned} Y_l &= \min_i(f(P_i)) \\ Y_h &= \max_i(f(P_i)). \end{aligned}$$

In the NM search strategy, the reflection (P_r), the expansion (P_e) and the contraction (P_c) transformations and centroid $\overline{P}$ are defined as follows:

$$\begin{aligned}
\overline{P} &= \frac{1}{n}\sum_{i=1}^{n} P_i, \\
P_r &= (1+\alpha)\overline{P} - \alpha P_h, \\
P_e &= \gamma P_r + (1-\gamma)\overline{P}, \\
P_c &= \beta P_h + (1-\beta)\overline{P},
\end{aligned}$$

where α, β and γ assume values in $[0,1]$ which change the volume of the simplex by reflection, contraction and expansion, respectively [43].

The algorithm of the NM search strategy is given in Algorithm 2.3 (taken from [2]).

Algorithm 2.3 Nelder-Mead search strategy (taken from [2])

```
 1: while stopping condition has not been reached do
 2:    Perform reflection for the point $P_h$,
 3:    Copy $P_r$ into $P_{temp}$,
 4:    if $f(P_{temp}) \leq Y_l$ then
 5:       Substitute $P_l$ with $P_{temp}$ and $Y_l$ with $f(P_{temp})$;
 6:    else
 7:       GOTO 12;
 8:    end if
 9:    Perform expansion at point $Y_l$,
10:    Copy $P_e$ into $P_{temp}$
11:    GOTO 4,
12:    if $(f(P_{temp}) < Y_h$ and $f(P_{temp}) > Y_i, i \neq h)$ then
13:       Replace $P_h$ by $P_{temp}$ and $f(P_h)$ by $f(P_{temp})$;
14:       Perform contraction of $P_h$;
15:       Copy $P_c$ into $P_{temp}$;
16:    end if
17:    if $f(P_{temp}) > Y_h$ then
18:       Replace all $P_i$'s by $\frac{(P_i+P_l)}{2}$;
19:    end if
20: end while
21: return $P_l$ and $Y_l$
```

In Algorithm 2.3, the stopping criterion can be set as in (2.1).

$$\frac{1}{n}\sum_{i=1}^{n} \| P_i^k - P_i^{k+1} \|^2 < \epsilon, \tag{2.1}$$

where P_i^k and P_i^{k+1} are the points in iteration k and $k+1$ respectively and ϵ is a very small real number [24].

2.3.2.2 Pattern search

Hooke-Jeeves is a pattern search optimization method [42]. In general a pattern search method does not use explicit derivatives. It first defines a pattern of points by moving each parameter one by one. So, for an optimization problem with n dimensions, it invokes a pattern containing at least $n+1$ points in each iteration by step size. The entire pattern of points is then shifted or moved to a new location. This new location is determined by selecting the best of each of the moves in the n parameters. The step sizes constantly become smaller so that the algorithm finds a good quality solution. To reduce the number of function evaluations, some pattern search methods evaluate fewer than $n+1$ functions in each evaluation [16]. The basic form of all the pattern search is given in Algorithm 2.4 [32].

Algorithm 2.4 Basic pattern search algorithm

```
while stopping condition has not been reached do
    Start from a point in the search space.
    Generate some neighbors around the current point and pick the one with the
    best objective function value.
    Depending on the how good the point is, change the step size
end while
```

2.3.3 Surrogate models

Surrogate models aim at modeling the underlying function of the objective function. High-fidelity surrogate models provide the gradient and higher order derivative information [31]. Surrogate models are generated by sampling the original objective function. The quality of the surrogate model is expected to improve with more sampling [32].

A very basic form of surrogate models was proposed by Schonlau as given in Algorithm 2.5 [47].

Algorithm 2.5 Basic surrogate model proposed in [47]

```
Evaluate few samples of the search space of the objective function
while stopping criterion is not meet do
    Estimate a function which fits all the evaluated points
    Find the minimum of the estimated function and use it as the next sample point
    Evaluate the new sample point
end while
return the minimum value
```

By having a larger set of evaluated points, the estimated function becomes closer to the objective function and, hence, the sought minimum value is expected to have a smaller value. For the sampling, Schonlau [47] applied Latin hypercube sampling schemes [60] as they have a space filling property which uniformly cover the search space. Schonlau proposed sampling of 10 points per each active variable [47]. For fitting the sampled data; polynomial, non-parametric regressions or etc. could be applied. The algorithm performs sampling until the improvement is smaller than a pre-defined threshold for a number of consecutive iterations.

2.4 Hybridization methods in meta-heuristics based approaches

There are a great deal of works on GCO using meta-heuristic approaches, particularly TS. In an early work, Siarry and Berthiau [61] applied Continuous TS (CTS) for GCOPs. They introduced a crown notation for *neighbour-search strategy*. So, instead of generating purely random neighbours around the current solution which makes the neighbours' relative location inhomogeneous, the space around the current solution is partitioned into crowns. This method suffered from low convergence speed, since it did not have any specifically designed phases for diversification and intensification. In another attempt to increase the speed of convergence, TS is divided into two phases [62]. In the first phase, the most promising areas are defined and then in the second phase the promising areas are intensified. In another recent work a three-phase TS approach has been proposed [22]. In the first phase of this approach the promising areas are recognized. The second phase investigates which of the promising areas has the highest potential. Then in the third phase, the potential area is intensified to find the global optimum solution.

More recent works are mostly based on hybrid methods. It is observed that some meta-heuristic methods are not efficient enough for local optimization. A hybrid method combining TS and Nelder-Mead (NM) has been proposed by Chelouah and Siarry [24]. In this method TS is applied for diversification and it returns the promising areas. Then the intensification is carried out on the promising areas by NM. Hedar and Fukushima [26] proposed almost a similar framework with Adaptive Pattern Search (APS) in the intensification phase.

TS is not the only meta-heuristic approach which is applied for GCO. Other meta-heuristic approaches like Simulated Annealing (SA) [49], Ant Colony Optimization (ACO) [27, 63, 64], Genetic Algorithms (GA) [51, 65, 66], Particle Swarm Optimization (PSO) [27, 65, 67–69] and harmony search [56, 70] have been applied in GCO. As mentioned earlier, meta-heuristic methods are mostly hybridized to increase the speed of convergence. It should be mentioned that the majority of the above mentioned approaches are hybridized with LOMs for the sake of increasing speed.

The existing methods can still be improved so that they can return a more accurate result in less time without being trapped in local minima. The basic TS is an efficient method to search for potential near optimum solutions in a large search space. It keeps track of the explored areas in a list which is called Tabu List (TL) and it prevents searching them in the near future. Thus considering the TL, the algorithm pushes the next moves to unexplored areas. Thus, there is a lower possibility to trap into local optimum solutions or to be stuck in some specific areas [55]. This makes TS a very effective technique for unknown optimization problems which may have many local optimum solutions. In this research, to increase the robustness of the proposed algorithm, we add a new step, called partitioning phase, prior to the diversification which is performed by TS. To increase the diversification speed while not being trapped into local minima, we improve the TS method. The improvement is done by proposing a new neighbour-search strategy. The original NM is applied at the end to intensify the near optimum solution and return the global optimum solution. As mentioned earlier, Global Optimizer Methods (GOM) use up a lot of time to find a very close solution in GCOPs if they are used alone. On the other hand, LOMs are very fast to exploit a local area but they cannot be applied for global optimization. This is the main motivation on hybridization of GOMs and LOMs for GCOPs.

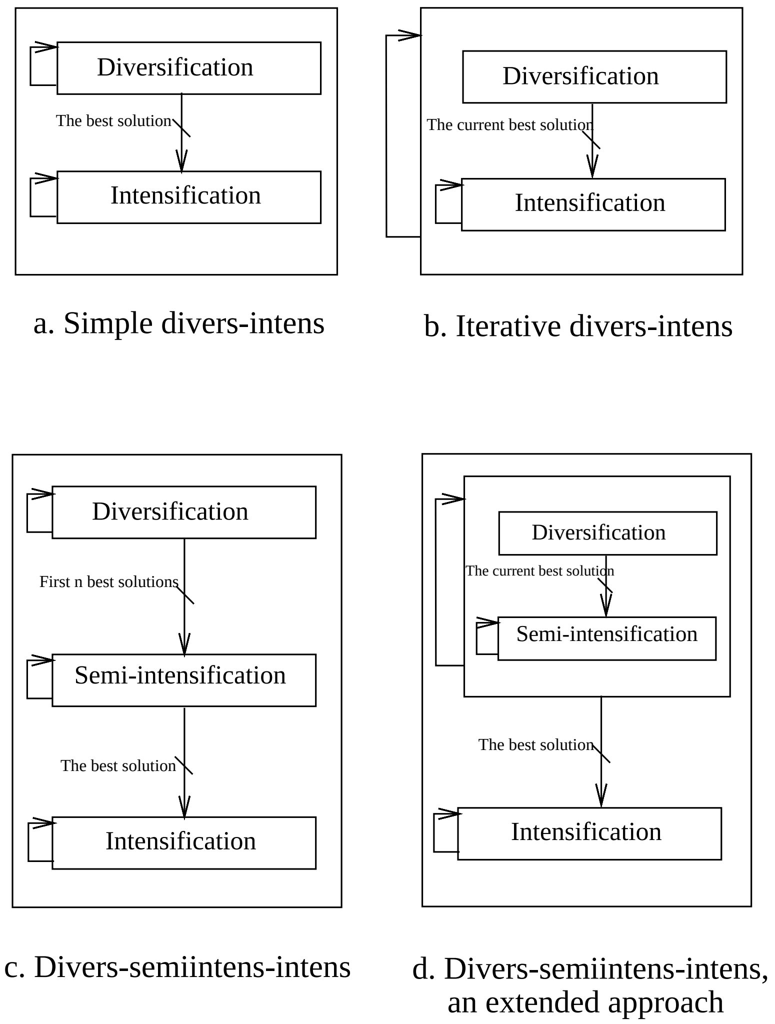
There are different ways of hybridazation of meta-heuristics as GOMs with LOMs; for a comprehensive study one can refer to [71]. The traditional hybridization of the most of GCOMs is based on diversification phase followed by the intensification. Both phases are computationally expensive. To decrease the computational cost, the diversification phase can stop earlier in order to start the intensification phase. However, the switching time between the diversification and the intensification is very important [49]. A very early switching to the intensification, increases the possibility of trapping in local minima. Conversely, a very late switch makes the diversification phase very expensive.

The most common meta-heuristic approaches which have been used in GCO are TS [22, 24–26, 29, 54, 61, 62, 72–74], GA [51, 66, 67], PSO [27, 67, 69, 75, 76], SA [49], artificial bee colony [77], artificial immune systems [78] and ACO [63, 64, 79–81]. For the intensification, different approaches like NM [24, 26, 51, 67, 76], proximal bundle method [49], APS [16, 26], Hooke-Jeeves direct search method [82], harmony search [6] and ACO [27] have been applied. However, some studies have improved the number of meta-heuristics such that they can be applied alone and perform both diversification and intensification. As an example, in an early work, dynamic DF was proposed for TS. Initially TS is applied with a large DF for the sake of diversification and as the algorithm is unable to find better solutions, the DF is halved [61]. As the DF gets smaller, the TS carries the intensification. On the other hand, some meta-heuristics, e.g. SA, naturally carry out both diversification and intensification. Diversification is done when the temperature factor is high and as the internal energy is reduced, the algorithm starts the intensification [49].

2.4.1 Hybridization frameworks in meta-heuristics

The diversification and intensification phases can be realized in different ways. Below are the main approaches of hybridization in GCO. Figure 2.1 shows an outline of these different hybridization strategies.

The simplest way of hybridization of a global optimizer and a local optimizer is to perform the diversification for a number of iterations and find the *"best-solution"* . Then a LOM starts the intensification from the best solution found in the diversification phase. In this case of hybridization, GOM stores the potential solution -whenever it finds one- in the *best-solution* and then starts searching again. If it finds another potential solution, it compares it with the current *best-solution*. If it is better, then it replaces the *best-solution* with the current potential solution. The



a. Simple divers-intens

b. Iterative divers-intens

c. Divers-semiintens-intens

d. Divers-semiintens-intens, an extended approach

Figure 2.1: Hybridization strategies for diversification and intensification (taken from [2])

process continues until it cannot find any potential solution which outperforms the *best-solution*. The *best-solution* is then used as the starting point of LOM. The LOM perfoms the intensification on the *best-solution* and return the global optimum solution at the end. We refer to this strategy as *"simple divers-intens"*, see Figure 2.1.a.

Another strategy of hybridization is to perform local optimization for every single potential solution. Thus whenever the algorithm finds a potential solution, it performs intensification on it and saves the result into the *best-solution*. Then it starts diversification from another unseen area for the sake of finding a new potential solution. Then again it does intensification on the new potential solution and compares the new obtained solution with the *best-solution*. If it is better, the algorithm replaces the *best-solution* with the obtained one. The algorithm continues until it cannot find any more new potential solutions in the diversification phase. We refer to this strategy as *"iterative divers-intens"*, Figure 2.1.b.

An optimized way of hybridization is to do the diversification phase and store all of the potential solutions in a list, known as *"promising-list"*. Then perform a semi-intensification phase on the stored potential solutions and keep the one with the minimum value in the *best-solution*. The obtained *best-solution* is used as the starting point of the intensification phase. We refer to this strategy as *"divers-semiintens-intens"*, Figure 2.1.c. There are different varieties of *divers-semiintens-intens* strategies in the literature. For example the diversification and the semi-intensfication can be performed in an *iterative divers-intens* manner. Then the best first n solutions are stored in the *promising-list* following by the intensification phase which performs local optimization to find the best solution, Figure 2.1.d.

The *simple divers-intens* strategy is very fast when compared to other strategies, but since only one best solution is considered for the intensification phase, it does not guarantee finding the optimum solution. As an example, suppose there are two potential solutions in the diversification phase; one is chosen as the *best-solution* that has less output value. But the *best-solution* does not necessarily show the best area since the other solution might have been around the global optimum solution but just could not obtain a lower output value. Figure 2.2 shows a typical example of this case.

To solve the problem of *simple divers-intens* hybridization strategy, *iterative divers-intens* can be applied. Since in this strategy every single potential found area is intensified, the possibility of finding the global optimum solution is increased. But this strategy is computationally very expensive since the intensifcation is carried out for all of the potential areas. When we find a potential area it might seem very promising but as we do diversification in future, we may find far better areas than the previous promising areas which apparently should not be intensified. Figure 2.3 shows an example of this case where it is computationally expensive.

To solve the computational issue of the *iterative divers-intens* while obtaining an acceptable rate of finding the global optimum solution, the *divers-semiintens-intens* hybridization strategy can be applied. In this strategy, the diversification stores only the best n promising areas in the *promising-list*. So we do not care about the other promising areas which are found in the diversification phase. Then a semi-intensfication phase is carried out on these n areas to see which one is the most promising one. Note that the first best promising area out of those n areas is not necessarily the one which is in the global optimal area. That is why we need to do

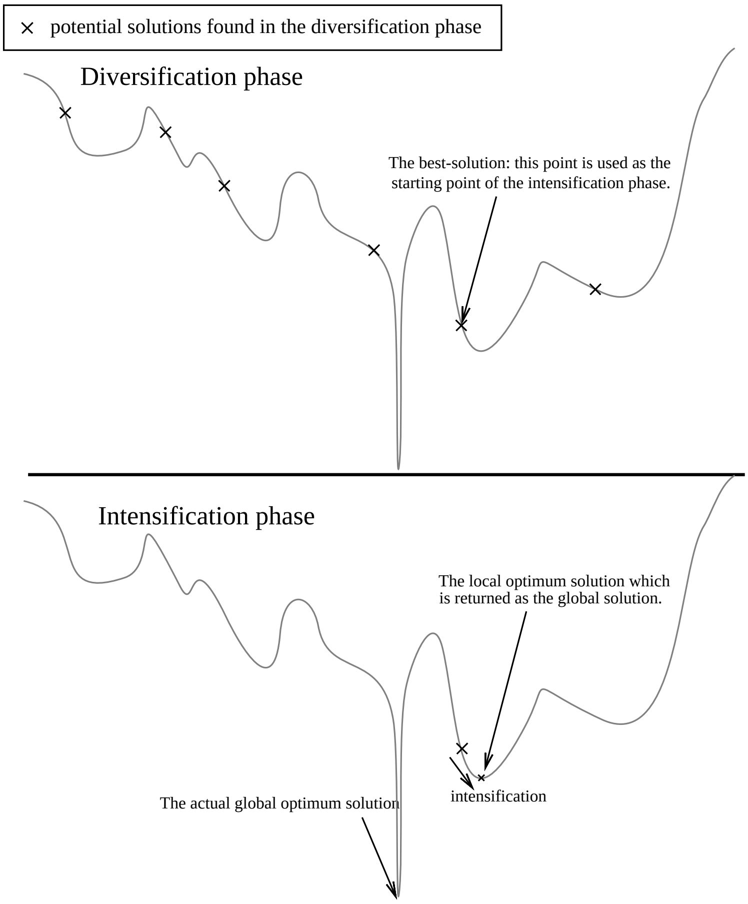

× potential solutions found in the diversification phase
Diversification phase
The best-solution: this point is used as the starting point of the intensification phase.
Intensification phase
The local optimum solution which is returned as the global solution.
intensification
The actual global optimum solution

Figure 2.2: An example of the *simple divers-intens* drawback (taken from [2])

the semi-intensfication phase. After semi-intensification the best promising area is given to the intensification phase for local optimization. Figure 2.4 shows this case with an example. In Figure 2.4, we assume n is equal to 3. Thus only the first three promising areas are considered during the semi-intensification phase.

2.5 Application of global optimization for fuzzy regression

One of the major benefits that one could get from the application of meta-heuristic approaches is that they are context-independent. So, a fitness measure to be minimized can be treated as a black-box. As mentioned earlier, one of the challenges in fuzzy linear regression is that the fitness measure can be a trade secret which cannot be transparent to the fuzzy linear regression analyst [30]. Another difficulty with the existing fuzzy linear regression analysis is that the mathematician is limited in

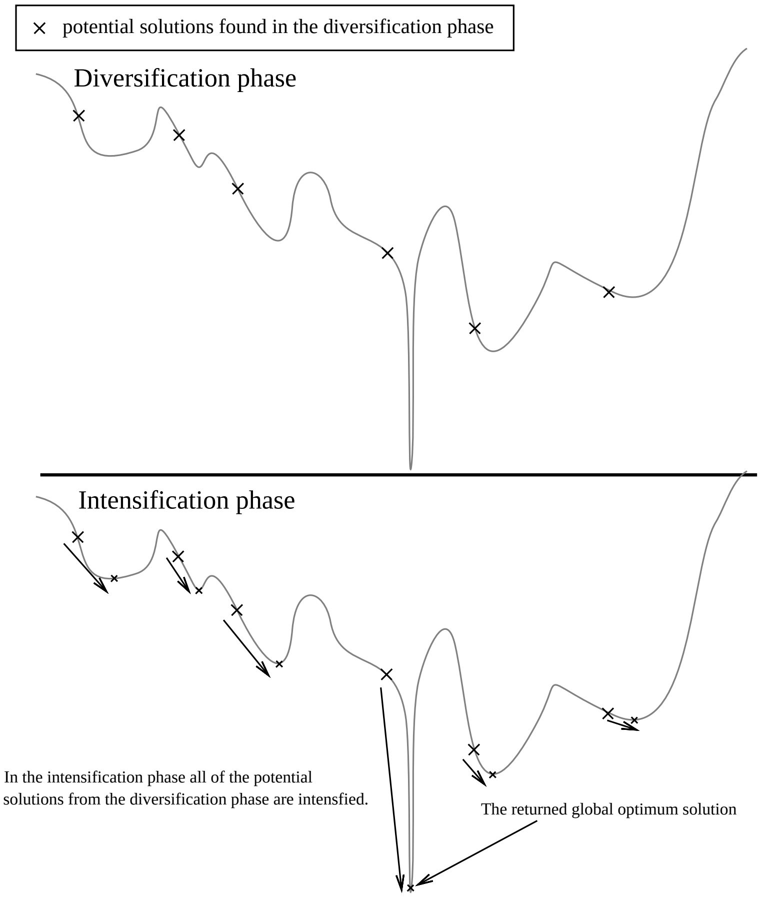

× potential solutions found in the diversification phase
Diversification phase
Intensification phase
In the intensification phase all of the potential solutions from the diversification phase are intensfied.
The returned global optimum solution

Figure 2.3: The computationally expensive example for *iterative divers-intens* hybridization strategy (taken from [2])

selecting the fitness measure. For example in least square approaches the fitness measure needs to be differentiable and thus must be continuous. The application of meta-heuristic approaches relaxes this restrictions as these approaches do not dictate any condition for the selection of the objective function [2]. So, the objective function can be designed in such a way that it can tackle the issue of dealing with outliers ([8]) and the spread increasing problem. Although there are other approaches to deal with outlier detection ([7, 83–85]) and the spread increasing problem ([86–88]) in the literature where global continuous optimization is not used in all of them the fitness measures are selected (or tweaked) in such a way that the least square or linear programming approaches could minimize the measure.

As illustrated in Figure 2.1, there are four major hybridization approaches. Depending on the application at hand,one may choose any of the approaches to increase the speed, accuracy, or to balance between both speed and accuracy. To this end, we assumed that for fuzzy linear regression analysis, we want to have a balance

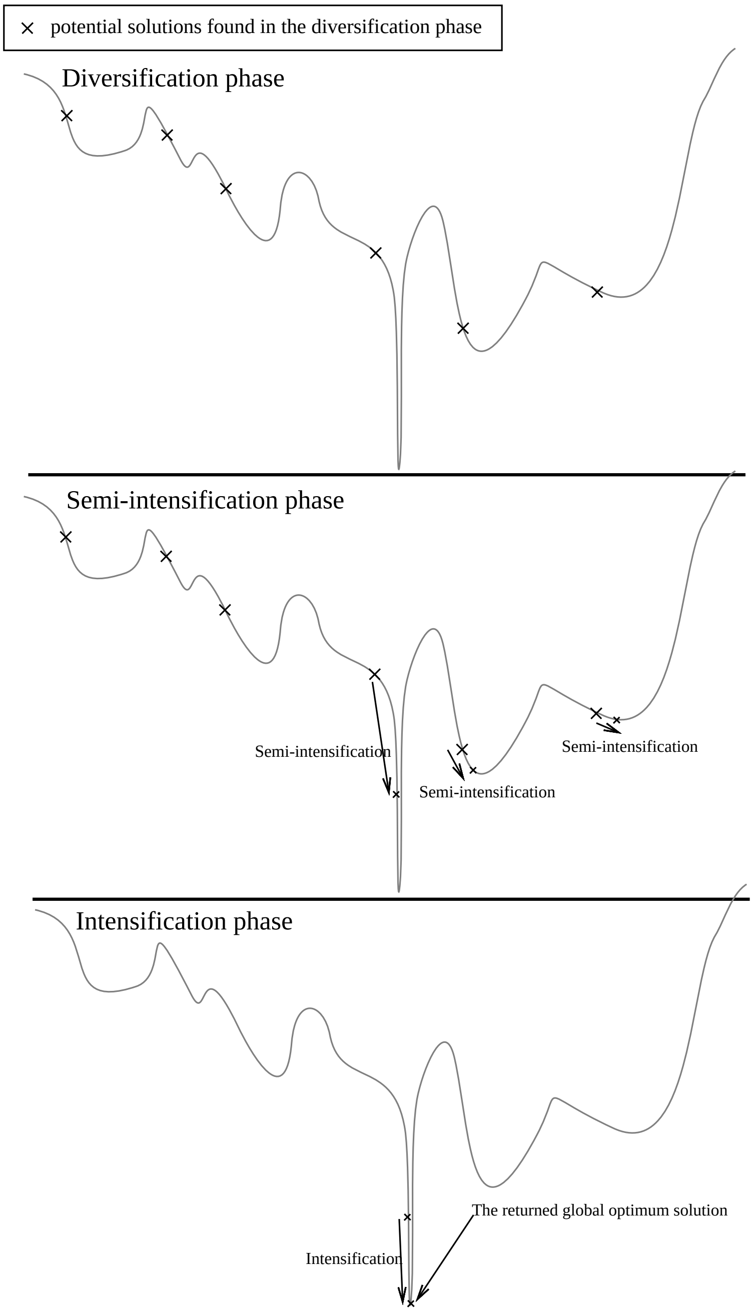

× potential solutions found in the diversification phase
Diversification phase
Semi-intensification phase
Semi-intensification
Semi-intensification
Semi-intensification
Intensification phase
The returned global optimum solution
Intensification

Figure 2.4: An example for *divers-semiintens-intens* hybridization strategy (taken from [2])

between the speed and accuracy (however if for an application this assumption is not valid, one could use another hybridization approach). In a preliminary study, we verified the *"divers-semiintens-intens"* for some benchmark functions and an industrial application [2]. The results are presented in Appendix A.1 and the details of the benchmark functions are given in Appendix A.2.

2.6 Summary

Global continuous optimization is a preferred option when the optimization problem at hand has either *unreliable derivatives*, or *black-box nature* and also satisfies any of the following properties *computationally expensive evaluation, costly or dangerous evaluations, optimization with time constraints*. Some applications that are classified into global continuous optimization include minimization of helicopter rotor vibration [30, 37, 38], and some chemistry and biology applications [30].

Three main approaches to deal with such optimization problems are *direct search methods, meta-heuristic based approaches* and *surrogate models*. Perhaps the oldest approaches are direct search methods which have a history of about half a century. These algorithms are mainly deterministic. Nelder-Mead and Hooke-Jeeves methods are discussed as an example of such methods. Another class of approaches for solving such optimization problems is *meta-heuristic based approaches*. Metaheuristics are stochastic methods which seek a good solution comprising some randomness. As an example, this chapter provides basic details of four meta-heuristics; genetic algorithms, tabu search, simulated annealing and harmony search. In surrogate models, as the last class of approaches for solving global continuous optimization problems, a surrogate model of the objective function is sought. If the high fidelity surrogate model is available then instead of studying the objective function itself to find the minimum point, the surrogate model can be investigated. The benefit of applying surrogate models is that they do not have *computationally expensive evaluation*, or *costly or dangerous evaluations*.

A slow rate of convergence is a drawback in some of meta-heuristic based approaches [2]. Hybridization is a solution toward solving this issue. Four major hybrid frameworks are described where each is suitable for a types of problems. It is reported that the *simple divers-intens* hybridization framework is very fast when compared to other frameworks, but since only one best solution is considered for the intensification phase, it does not guarantee finding the optimum solution. On the other hand, *iterative divers-intens* hybridization frameworks increase the possibility of finding the global optimum solution. But this framework is computationally very expensive since intensification is carried out for all of the potential areas. *Divers-semiintens-intens* frameworks are realized to keep the balance between the computational cost and the rate of finding global optimum solutions.

3 Fuzzy linear regression[1]

Fuzzy linear regression analysis is a powerful tool to model the input-output relationship and for forecasting purposes or studying the behavior of the data. The existing challenges in fuzzy linear regression are, dealing with non-transparent fitness measures, outlier detection and the spread increasing problem. This chapter studies the existing methods and proposes the application of the global continuous optimization to tackle these issues.

3.1 Fuzzy logic

Many real-world problems cannot be expressed by binary logic. In particular most of human inferences are not consistent with the two-valued logic [90]. For example the expressions such as; *comfortability* and *cloudy* in the sentences "Is the weather cloudy?" and "Is the temperature comfortable?" cannot be suitably described by binary logic [3, 90]. The meaning of the *comfortability* is far more than something that can be represented by one number. Interval variables can be used to represent such variables, for example a temperature is comfortable if it is in a closed interval

[1] This chapter is based on a journal paper; [6] co-authored by M. A. Orgun, M. Mashinchi and W. Pedrycz, and a conference paper; [89] co-authored by M. A. Orgun and M. Mashinchi

of $[20°, 25°]$, where the universe is $-20°$ to $50°$. However, the interval representation treats all the temperatures in the interval as *comfortable*. A more generalized way is to give membership degrees to each of the values in this interval. Assuming that the temperature is perfectly comfortable at $22°$ to $23°$ and the satisfaction feeling decreases with lower or higher values, then the fuzzy set for *comfortability* can be defined as given in Figure 3.1 [3].

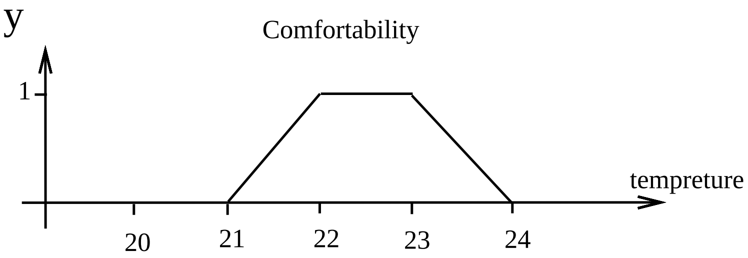


Figure 3.1: The fuzzy set of the comfortability of the temperature (derived from [3])

3.1.1 Fuzzy sets and fuzzy numbers

A fuzzy set is a "class of objects with a continuum grades of memberships" [91] where a membership value can be any value from zero to one and defines how much a member belongs to that set [92]. A fuzzy set $\widetilde{A}$ which is subset of X is formally defined as given in 3.1.

$$\widetilde{A} = \{(x, \mu_A(x)) | x \epsilon X\} \tag{3.1}$$

where $\mu_A(x)$ is the grade of membership of x in $\widetilde{A}$. Value of $\mu_A(x)$ is in the closed interval of $[0,1]$. The closer $\mu_A(x)$ is to 1, the greater is the degree to which x belongs to $\widetilde{A}$. Similarly the closer $\mu_A(x)$ is to 0, the less x belongs to $\widetilde{A}$ [93, 94]. If the interval [0,1] is substituted with the binary set $\{0,1\}$, then the set A is converted to a regular set [94].

Some other concepts of fuzzy sets are defined as follows:

Definition 1 ($\alpha-$level)**.** *For $\alpha\epsilon[0,1]$, the $\alpha-$level set of a fuzzy set $\widetilde{A}$ is the crisp set:*

$$\widetilde{A}_\alpha = \{x \epsilon X | \widetilde{A}(x) \geq \alpha\}. \tag{3.2}$$

The interval representation of $\alpha-$level can be defined as $\widetilde{A}_\alpha = [A^l_\alpha, A^r_\alpha]$.

Definition 2 (Support). *The support of a fuzzy set $\widetilde{A}$ on X, denoted by $supp(\widetilde{A})$, is the set of points in X at which $\widetilde{A}(x) > 0$;*

$$supp(\widetilde{A}) = \{x \epsilon X | \widetilde{A}(x) > 0\}. \tag{3.3}$$

Definition 3 (Height). *The height of a fuzzy set $\widetilde{A}$ on X, denoted by $hgt(\widetilde{A})$, is the least upper sup() bound of $\widetilde{A}(x)$;*

$$hgt(\widetilde{A}) = \sup_{x \epsilon X} \widetilde{A}(x) \tag{3.4}$$

Definition 4 (Convexity). *A fuzzy set $\widetilde{A}$ is said to be convex if for any $\alpha \epsilon [0,1]$, $\widetilde{A}_\alpha$ is a convex set.*

Definition 5 (Fuzzy number). *Fuzzy numbers are special kinds of fuzzy sets. A fuzzy number is a convex normal fuzzy set of the real line $X = \mathbb{R}^1$ whose membership function is piecewise continuous.*

An arbitrary fuzzy number $\widetilde{A}$ can be represented by an ordered pair of continuous functions $[A^l(\alpha), A^r(\alpha)]$ for $0 \leq \alpha \leq 1$ that satisfy the followings:

1. $A^l(\alpha)$ is increasing on $[0,1]$,
2. $A^r(\alpha)$ is decreasing on $[0,1]$,
3. $A^l(\alpha) \leq A^r(\alpha)$.

Definition 6 (Positive fuzzy number). *A fuzzy number $\widetilde{A}$ is called positive (non-negative), denoted by $\widetilde{A} > 0$, if $\forall x \leq 0$, its membership function $\widetilde{A} = 0$. A negative (non-positive) fuzzy number is defined similarly.*

Definition 7 (L-R fuzzy number). *A fuzzy number $\widetilde{A}$ is said to be an L-R fuzzy number if;*

$$\widetilde{A}_{(x)} = \begin{cases} L(\frac{a-x}{\alpha}) & x \leq 0, \alpha > 0 \\ L(\frac{x-a}{\beta}) & x > 0, \beta > 0 \end{cases}$$

where a is the mean value of $\widetilde{A}$ and α and β are left and right spreads, respectively, and L() is a left shape function satisfying;

- *$L(x) = L(-x)$,*
- *$L(0) = 1$, and*
- *$L(x)$ is nonincreasing on $[0,1)$.*

$R()$ is defined similarly to $L()$.

Such an L-R fuzzy number is denoted by $\widetilde{A} = (a;\alpha;\beta)_{LR}$, by using its mean value (a), left and right spread functions α and β. If an L-R fuzzy number is symmetric then it can be represented by $\widetilde{A} = (a;\alpha)$ where α is the left and right spreads of the fuzzy number.

3.1.2 Fuzzy arithmetics

Fuzzy arithmetics which are defined by extension principle provides necessary operations for the manipulation of fuzzy numbers [3].

Definition 8 (Extension principle). *Let $f : X \longrightarrow Y$ be a mapping function from a set X to a set Y. For each fuzzy set $\widetilde{A}$ in X the fuzzy set $\widetilde{B}$ in Y is induced by f as follows:*

$$\widetilde{B} = \{(y, \widetilde{B}(y)) | y = f(x), x \epsilon X\} \ with \tag{3.5}$$
$$\widetilde{B}(y) = \begin{cases} \sup_{y=f(x), x\epsilon X} \widetilde{A}(x) & f^{-1}(y) \neq \phi \\ 0 & f^{-1}(y) = \phi \end{cases}$$

where, $f^{-1}(y)$ is the inverse image of y.

Now, by setting $X = \mathbb{R}^2$ and $Y = \mathbb{R}$, the addition, subtraction, and multiplication of two fuzzy numbers $\widetilde{A}$ and $\widetilde{B}$ are obtained as follows [90]:

Addition:

$$(\widetilde{A} + \widetilde{B})(x) = \sup_{x=a+b} \min\{\widetilde{A}(a), \widetilde{(B)}(b)\} = \sup_{a\epsilon\mathbb{R}} \min\{\widetilde{A}(a), \widetilde{(B)}(x-a)\}$$

Subtraction:

$$(\widetilde{A} - \widetilde{B})(x) = \sup_{x=a-b} \min\{\widetilde{A}(a), \widetilde{(B)}(b)\} = \sup_{a\epsilon\mathbb{R}} \min\{\widetilde{A}(a), \widetilde{(B)}(x-a)\}$$

Multiplication:

$$(\widetilde{A} \times \widetilde{B}) \sup_{x=ab} \min\{\widetilde{A}(a), \widetilde{(B)}(b)\}$$

For L-R fuzzy numbers such as $\widetilde{A} = (a, \alpha, \beta)_{LR}$ and $\widetilde{B} = (b, \lambda, \sigma)_{LR}$, the addition, subtraction and approximate formulas for multiplication are defined as follows [95].

Addition:

$$(\widetilde{A} + \widetilde{B}) = (a + b, \alpha + \lambda, \beta + \sigma)_{LR}$$

Subtraction:

$$(\widetilde{A} - \widetilde{B}) = (a - b, \alpha + \lambda, \beta + \sigma)_{LR}$$

Multiplication:

$$(\widetilde{A} \times \widetilde{B}) = \begin{cases} (ab, a\lambda + b\alpha, a\sigma + b\beta)_{LR} & \widetilde{A} > 0, \widetilde{B} > 0 \\ (ab, b\alpha - a\sigma, b\beta + a\sigma)_{LR} & \widetilde{A} \leq 0, \widetilde{B} > 0 \\ (ab, -b\beta - a\sigma, -b\alpha - a\sigma)_{LR} & \widetilde{A} < 0, \widetilde{B} \leq 0 \end{cases}$$

3.1.3 Fuzzy similarity measures

The evaluation of how well a model can fit the observed data is done by a measurement which is called a fitness measure [92]. A distance-based fitness measure for fuzzy numbers is often generalized from a distance measure for interval numbers. Any concept of distance must satisfy the properties in the following definition;

Definition 9 (Property of a distance measure [93]). *A numerical function $D(A, B) \epsilon \mathbb{R}$ which is defined on a set $\mathbb{E}$ is a distance if and only if:*
$\forall A, B, C \epsilon \mathbb{E}:$

$$\begin{aligned} &D(A, B) \geq 0, \\ &D(A, B) = D(B, A), \\ &D(A, B) = 0 \Rightarrow A = B, \\ &D(A, C) \leq D(A, B) + D(B, C). \end{aligned}$$

Kaufman and Gupta [96] defined the distance measure for two fuzzy numbers $\widetilde{A}$ and $\widetilde{B}$ as follows:

$$D(A, B) = \int_{\alpha=0}^{\alpha=1} |A_{\alpha}^{l} - B_{\alpha}^{l}| + |A_{\alpha}^{r} - B_{\alpha}^{r}| d\alpha$$

.

Kaufman and Gupta's measure is the generalized form of the distance measure proposed by Dimond [97] and Bardossy et al. [98] for two interval numbers $A =$

(a_1, a_2) and $B = (b_1, b_2)$ as follows:

$$D(A,B) = \sqrt{(a_2 - b_2)^2 + (a_1 - b_1)^2}.$$

Dimond [97] defined a distance measure for two fuzzy numbers $\widetilde{A} = (a, \alpha, \beta)$ and $\widetilde{B} = (b, \lambda, \sigma)$ as follows:

$$D = (\widetilde{A}, \widetilde{B}) = (a-b)^2 + (a - \alpha - b + \lambda)^2 + (a - \beta - b + \sigma)^2$$

In case A and B are crisp values, Dimonds measure is equivalent to triple of the Euclidean distance. The measure needs to be modified to handle trapezoidal fuzzy numbers [93].

There are other model fitting measurements (MFMs) such as Hojati's similarity measure [4], distance criterion [99], Euclidean distance [100], none-intersected area [88,101], relative none-intersected area [12], and compatibility measure [102]. Since there is no evidence that one measure is better than the others, one may adopt a MFM based upon requirements of modelling and the nature of the data set. Followings are three more similarity measures proposed by Hojati [4] given by equation (6.3), distance criterion which is proposed in [99] expressed as equation (4.8), and the none-intersected area (NIA) proposed in [88,101] and defined by equation (4.9). The numerator of equation (6.3), the none-intersected area, for two fuzzy numbers is illustrated in Figure (3.2).

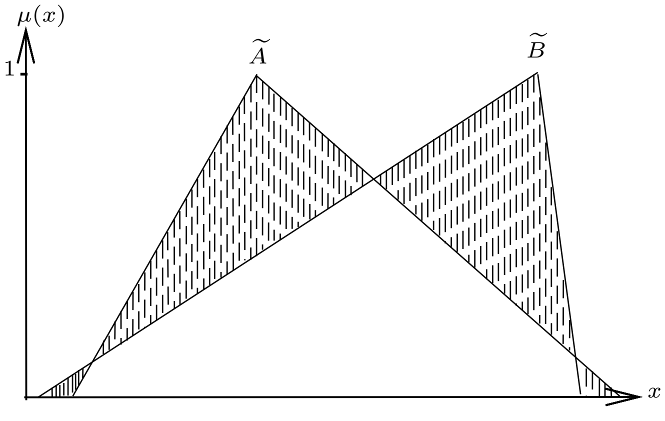


Figure 3.2: An example of the non-intersected area of two fuzzy numbers which is used in the numerator of the objective function given in equation (6.3) (derived from [4])

$$S_h = \frac{\int \min(\mu_{\widetilde{y}^*(x)}, \mu_{\widetilde{y}(x)})dx}{\int \max(\mu_{\widetilde{y}^*(x)}, \mu_{\widetilde{y}(x)})dx} \tag{3.6}$$

$$D = \frac{1}{4}\sum_{i=1}^{k} |y_i^L - y_i^{*L}| + 2|y_i^C - y_i^{*C}| + |y_i^R - y_i^{*R}| \tag{3.7}$$

$$NIA = \sum_{i=1}^{k}\left[\int_{S_{\widetilde{y}_i^*} \cup S_{\widetilde{y}_i}} |\mu_{\widetilde{y}_i(x)} - \mu_{\widetilde{y}_i^*(x)}| dx\right]. \tag{3.8}$$

.

3.2 Fuzzy regression

Classical regression analysis offers a conceptual and algorithmic vehicle to discover relationships (functions) between independent (explanatory, covariant, input) variables and dependent (response, output, model's estimated output) variables [103, 104]. The problem is to optimize the given function's parameters for the given input-output data so that a predetermined objective function attains its (global) minimum [88]. The ultimate objective of forming a regression model is to estimate the value of a continuous dependent variable for any arbitrary value of the independent variable [105]. Regression models are important tools in operations research, complex systems analysis and various fields of application; such as economy, finance, marketing, social sciences, healthcare and others [101,106].

Classical regression realized by means of statistical techniques is applied successfully to analyze quantitative data and homogeneous observations [104]. The deviation between the observed data and the estimated data encountered in classical regression is due to the measurement error or random variations of parameters [1,107,108]. Such random variations can be represented as a normal distribution of some variance and zero mean, which makes statistical techniques effective in determining the functional relationship for such types of data [107].

However, often a probabilistic representation of data is not available or not suitable, and there are significant deviations between the observed data and the corresponding estimates because of the imprecision introduced by human judgement or the indefinitness/vagueness of the model's structure [5,12]. In such systems, uncertainty arises not due to randomness but due to the phenomenon of fuzziness [108]. In "conventional" approaches, a numeric form of data is considered to construct the model – even if the data comes with some imprecision or uncertainty. As a result, this makes the estimated model not fully efficient since by considering the numeric form of data instead of uncertain data, some important information may have been overlooked or neglected [105].

The factor of uncertainty may emerge in the system's behaviour is due to several reasons:

- the high complexity of the environment, which necessiates the adaptation of abstraction (granulation of information) for generalization purposes [109],
- the influence of human subjective judgement in the decision process or the involvement of human-machine interactions [5, 12, 110], and
- partially available information [13], due to miss-recording or inaccurate measurements [111].

Quantization of uncertain data for simplication may cause partially available information [13]. In the quantization process some useful information are overlooked. Quality prediction based on explanatory variables is an example where the quality is usually represented by ordinal data [10].

The design of a regression model in such environments has been a challenge the design of classical regression. In classical aproaches, fuzzy data used to treated as ordinal data so that the classical statistical approaches can be applied [112]. As mentioned earlier, there is often useful information that can be overlooked at the defuzzification stage. An intuitively appealing approach towards the fuzziness of a system is to not defuzzify the data but to take the vagueness into consideration in the level of inferencing [112]. The defuzzification is only applied at the decision stage when it deemed to become necessary [112].

The representation of an experimental environment which is governed by uncertainty or impreciseness, can involve interval or fuzzy data [113]. To derive the corresponding models in such environments, Fuzzy Regression (FR) or interval regression (which is regarded as a simple version of FR [114]) is considered.

The FR model is referred to as a fuzzy or possibilistic model of classical regression, while classical regression is based on the principles of statistics [107]. It has been observed that the FR model is more effective than the classical regression when the normality of error terms and dependent variables, and the availability of a sufficiently large data set (complete data conditions) are not satisfied [5, 86, 115]. In such cases, the FR model may be used to explore the imprecise relationship between dependent and independent variables for the given system. The relationship is estimated by minimizing some error criterion capturing various facets of uncertainty [1, 116]. There have been many diverse applications of Fuzzy Linear Regression (FLR), such as R&D project evaluation [117], housing [5, 109], insurance [118] and many others.

In the FLR model, we are interested in finding a fuzzy function $\widetilde{y}$ in the form given below (3.9) which fits a finite number of numeric input-fuzzy output data with a minimum error [5], where the data is $\left((x_{1i}, x_{2i}, \cdots, x_{(n-1)i}, x_{ni}), \widetilde{y}_i^*\right)$ $i = 1, \cdots, k$.

$$\widetilde{y}_i = \widetilde{A}_n x_{ni} + \widetilde{A}_{n-1} x_{(n-1)i} + \cdots + \widetilde{A}_j x_{ji} + \cdots + \widetilde{A}_1 x_{1i} + \widetilde{A}_0 \tag{3.9}$$

In a more general situation, both inputs and outputs are treated as fuzzy numbers. In this case, we are looking for a function such as the one given by (3.10), to fit the fuzzy input-fuzzy output data $\left((\widetilde{x}_{1i}, \widetilde{x}_{2i}, \cdots, \widetilde{x}_{(n-1)i}, \widetilde{x}_{ni}), \widetilde{y}_i^*\right)$ [14].

$$\widetilde{y}_i = \widetilde{A}_n \widetilde{x}_{ni} + \widetilde{A}_{n-1} \widetilde{x}_{(n-1)i} + \cdots + \widetilde{A}_j \widetilde{x}_{ji} + \cdots + \widetilde{A}_1 \widetilde{x}_{1i} + \widetilde{A}_0 \tag{3.10}$$

where in (3.9) and (3.10), $\widetilde{A}_j$ $(j = 0, \cdots, n)$ is the fuzzy coefficient (parameter) of the regression model. The parameters are optimized in such a way that the differences between the observed outputs $\widetilde{y}_i^*$ and estimated ones $\widetilde{y}_i$ are made as small as possible. All $\widetilde{y}_i$ $(i = 1, \cdots, k)$, $\widetilde{A}_j$ $(j = 0, \cdots, n)$ and $\widetilde{x}_{ji}$ are fuzzy numbers and the operations used there are treated as the multiplication and addition of fuzzy numbers.

There are two approaches for FLRA; possibilistic and Least Square (LS) [1]. In the former approach proposed by Tanaka *et al.* [5], the aim is to minimize the whole fuzziness by minimizing the total spreads of the fuzzy coefficients while the estimated outputs and the observed ones are within a certain h-level of belief [107]. The LS-based approach proposed by Dimond [119] aims at minimizing the least square error between the estimated and the observed fuzzy data.

3.2.1 Linear programming approaches

Tanaka et al. [5] proposed a possibilistic approach which tries to minimize the fuzziness of the model by minimizing the spreads of the fuzzy coefficients where also each sample is captured within a feasible interval [1]. This model was proposed for crisp input-fuzzy output data based on linear programming where later on a more generalized model for fuzzy input-fuzzy output proposed [14]. The difference in the linear programming based approaches is based on their objective function to be minimized and their linear conditions.

The initial model investigated by Tanaka et al. [5] was proposed for symmetric triangular fuzzy numbers as follows:

$$\begin{aligned}
&\min \beta_0 + \beta_1 + \cdots + \beta_n \\
&\text{subject to: } \sum_{j=0}^{k} (\alpha_j + (1-h).\beta_j).x_{ij} \geq y_i + (1-h)e_i, i = 1, \cdots, n \\
&\sum_{j=0}^{k} (\alpha_j + (1-h).\beta_j).x_{ij} \leq y_i - (1-h)e_i, i = 1, \cdots, n \\
&\alpha_j \epsilon [0,1], \beta_j \leq 0
\end{aligned} \tag{3.11}$$

The solution to the linear program (3.11) forces the model to include the h-level of the observed interval to be included by the h-level of the predicted interval [4]. Figure (3.3) illustrates a model where the predicted intervals (dotted lines) include the observed intervals (bold line).

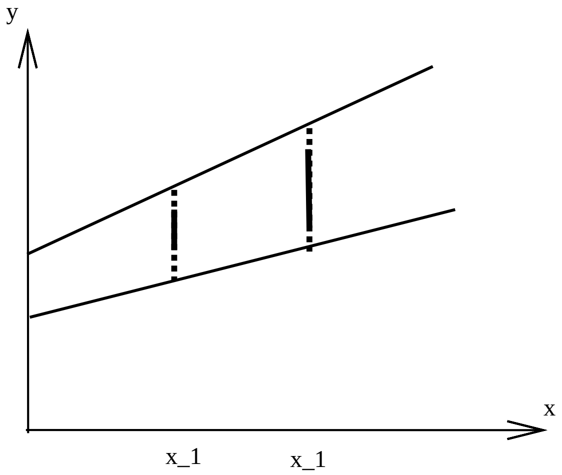


Figure 3.3: Illustration of Tanaka et al. model [5] (derived from [4])

Tanaka et al. [11], and Redden and Woodall [111] suggested to change the objective function to solve the shortcomings reported by [120] of the linear programming given in (3.11) where many β_i turn out to be zero and the models coefficients are independent of the input variables. The modified objective function is given in 3.12.

$$\min \beta_0 + \sum_{i=1}^{n} \sum_{j=0}^{k} \beta_i x_{ij} \tag{3.12}$$

Another major shortcoming of the Tanaka et al. model is that the h-level of the predicted interval is required to contain the corresponding observed interval, as illustrated in Figure 3.3. This results in obtaining a model where it is very sensitive to outliers and the data which have high imprecision (long spreads). A solution to this problem is given by allowing the model to intersect between the h-levels of predicted and observed intervals [11]. Peters modified the approach proposed by Tanaka et al. [5] to detect the outliers for crisp input and crisp output data by linear programming [85]. The linear program in Peters' approach maximizes the λ value which is in the interval $[0,1]$. The greater the λ value the better the quality of the data. However, in this approach, there are three user defined variables: d_0, p_i and p_0. The linear programming model is given as follows (3.13):

$$\begin{aligned}
&\max \lambda, \\
&\quad \text{subject to:} \\
&\quad (1-\lambda)p_0 - \sum_{i=1}^{N} c^t |x_i| \geq -d_0 \\
&\quad (1-\lambda)p_i + \alpha^t x_i + (1-h)c^t |x_i| \geq y_i \\
&\quad \alpha^t x_i + (1-h)c^t |x_i| - (1-\lambda)p_i \leq y_i \\
&\quad 0 \leq \lambda \leq 1, c \leq 0, i = 1, 2, \cdot, N
\end{aligned} \tag{3.13}$$

A small d_0 value means that we are interested in a function with smaller spreads. Peters actually suggested $d_0 = 0$ which means the best function is a crisp function which ideally is rational as both inputs and outputs are crisp. Also p_i and p_0 are user defined values which are context-dependent. For example, if a dataset is assumed to have a lot of outliers then one may use greater p_i and p_0 values.

Sakawa and Yano proposed a generalized model to deal with fuzzy input-fuzzy output data which is later on imporved by Hojati et al. [4]. Hojati set the objective function to the total sum of the distances of right and left spreads of predicted and observed data. It is noted that Hojati et al. approach does not work for asymmetric fuzzy numbers [4].

3.2.2 Least square approaches

Dimond [97,119] proposed another way of fuzzy linear regression analysis by solving an unconstrained least squares problem which yields solutions analogous to ordinary least squares. The formulation of least squares fuzzy linear regression is as follows:

$$\min \sum D^2(\widetilde{O}, \widetilde{P})$$

where $\widetilde{O}$ and $\widetilde{P}$ represent the observed and predicted fuzzy outputs, respectively. $D()$ is a distance measure between two fuzzy values.

The main idea in least square fuzzy linear regression analysis is to seek for the optimized coefficients in fuzzy linear regression which minimize the defined distance measure for observed and predicted data. Based on the chosen distance measure, and the type of the fuzzy input or output, different coefficients would be obtained. The coefficients for a model with one crisp input variable, one fuzzy triangular output variable with the distance measure (3.15) can be computed as follows [103]:

case 1: if $A_1 \geq 0$ in (3.9)

$$\begin{aligned} S^+(A_0, A_1) &= \sum_{i=1}^{n} D^(\widetilde{y_i}, A_0 + A_1\widetilde{x_i}) \\ &= \sum_{i=1}^{n} [(y_i - A_0 - A_1 x_i)^2 + (y_i^l - A_0 - A_1 x_i^l)^2 + (y_i^r - A_0 - A_1 x_i^r)^2] \end{aligned} \tag{3.14}$$

The coefficients A_0 and A_1 can be computed by differentiating the distance measure with respect to A_0 and A_1 as follows:

$$A_1 = \frac{\sum_{i=1}^{n} x_i^l y_i^l + x_i y_i + x_i^r y_i^r - 3n\bar{x}\bar{y}}{\sum_{i=1}^{n} (x_i^l)^2 + x_i)^2 + x_i^r)^2 - 3n\bar{x}^2}$$

$$A_0 = \bar{y} - A_1\bar{x}$$

where $\bar{y} = \frac{\sum_{i=1}^{n}(y_i^l + y_i + y_i^r)}{3n}$, and $\bar{x} = \frac{\sum_{i=1}^{n}(x_i^l + x_i + x_i^r)}{3n}$.

case 2: if $A_1 < 0$ in (3.9)

$$\begin{aligned} S^{+}(A_0,A_1) &= \sum_{i=1}^{n} D^{(}\widetilde{y}_i, A_0 + A_1\widetilde{x}_i) \\ &= \sum_{i=1}^{n} [(y_i - A_0 - A_1 x_i)^2 + (y_i^l - A_0 - A_1 x_i^r)^2 + (y_i^l - A_0 - A_1 x_i^r)^2] \end{aligned} \tag{3.15}$$

The coefficients; A_0 and A_1 can be computed by differentiating the distance measure with respect to A_0 and A_1 as follows:

$$A_1 = \frac{\sum_{i=1}^{n} x_i^l y_i^r + x_i y_i + x_i^l y_i^r - 3n\bar{x}\bar{y}}{\sum_{i=1}^{n} (x_i^l)^2 + x_i)^2 + x_i^r)^2 - 3n\bar{x}^2}$$

$$A_0 = \bar{y} - A_1\bar{x}$$

where $\bar{y} = \frac{\sum_{i=1}^{n}(y_i^l + y_i + y_i^r)}{3n}$, and $\bar{x} = \frac{\sum_{i=1}^{n}(x_i^l + x_i + x_i^r)}{3n}$.

The coefficients for a more generalized model with one fuzzy input variable, one fuzzy triangular output variable with the distance measure (3.16) can be computed as follows [103]:

$$\begin{aligned} S(A_0,A_1) &= \sum_{i=1}^{n} D^{(}\widetilde{y}_i, \widetilde{A}_0 + \widetilde{A}_1\widetilde{x}_i) \\ &= \sum_{i=1}^{n} [(y_i - a_0 - a_1 x_i)^2 + (y_i^l - a_0^l - a_1^l x_i^l)^2 + (y_i^r - a_0^r - a_1^r x_i^r)^2] \end{aligned} \tag{3.16}$$

Differentiating with respect to a_0, a_1, a_0^l, a_1^l, a_0^r, and a_1^r, the coefficients are as follows:

$$a_1^l = \frac{\sum_{i=1}^{n} x_i^l y_i^l - n\bar{x}^l \bar{y}^{l\,2}}{\sum_{i=1}^{n} (x_i^l)^2 - n\bar{x}^l}, \quad a_0^l = \bar{y}^l - a_1^l \bar{x}^l$$

$$a_1 = \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}^{2}}{\sum_{i=1}^{n} (x_i)^2 - n\bar{x}}, \quad a_0 = \bar{y} - a_1\bar{x}$$

$$a_1^r = \frac{\sum_{i=1}^{n} x_i^r y_i^r - n\bar{x}^r \bar{y}^{r\,2}}{\sum_{i=1}^{n} (x_i^r)^2 - n\bar{x}^r} \;, \quad a_0^r = \bar{y}^r - a_1^r \bar{x}^r$$

All the least square approaches in the literature share the above given concept in which the chosen distance measure obtains its minimum by differentiation with respect to the coefficients. However, what make these methods distinct are; defining an algebraic operation such as approximation of the product operation [121,122], an application of different a distance measure [123], a more generalized type of fuzzy coffiecients such as trapezoidal fuzzy numbers [124].

3.2.3 Comparison between least square and linear programming approaches

The first FLR model, proposed by Tanaka et al. [5], was based on the possibility theory and solved by Linear Programming (LP) [1, 5, 125]. Celmins [126] and Dimond [97] proposed a FR model based on the Least Square (LS) approach. According to the error definition, FR models are classified in two categories [1]; possibilistic with LP, and LS approaches. In the former approach, the aim is to minimize the overall fuzziness by minimizing the total spread of the fuzzy coefficients while the estimated outputs and the observed ones are within a certain h-level of confidence [86, 107]. The term h expresses the fitness between the estimated fuzzy outputs and the observed ones [107]. On the other hand, in the LS-based model, a similarity measure between observed and estimated outputs is used as the measurement for the fitness of the model [1,107].

LP is a well-known technique used in possibilistic approaches to minimize the fuzziness of the coefficients. In the LP based approaches with additional observation data, two constraints are added to LP [103]. However, having extra constraints is not an issue in practice since the data sets for fuzzy regression are small and also, LP problems even with thousands of variables and constraints can be solved in a few seconds. Tanaka et al.'s LP approach [5] is criticized to be very sensitive to outliers [86, 111, 123] and if more input-output data are provided, then the spreads of estimated outputs may become undesirably wide [86, 111, 123]. However considering the goal of LP based approaches which is to cover the spreads of all the observations (up to an h-level), the sensitivity of these approaches to outliers or having wider spreads with more observations cannot be counted as drawbacks. Furthermore, in LP based approaches there is a tunable parameter by which the level of uncertainty might be kept under control. On the other hand, the goal of LS-based approaches which are the extension of the statistical regression analysis differs from LP based approaches. The goal of LS based approaches is to find a model which has the most similar estimated outputs to the observed ones based on the chosen similarity measure.

3.3 Challenges in fuzzy linear regression

Perhaps the most important part in an optimization problem is the selection of the objective function. The sought model could be varied based on the chosen objective function. Fuzzy linear regression analysis can be seen as an optimization problem where the aim is to derive a model which it fits the given data set. Another challenge in fuzzy regression analysis is to obtain a model which is insensitive of the outliers. Although the portion of the outliers is usually small comparing to the rest of the data set, a model which fits all the data including the outliers will have an unpredictable behavior. The last issue in fuzzy linear regression analysis, which is introduced from the early stages of the proposal of fuzzy linear regression in 1982, is the spread increasing problem (SIP) in which the uncertainty of the model grows with the increase of the magnitude of the input.

3.3.1 Fitness measure

The selection of the fitness measure is one of the most important issues in fuzzy linear regression analysis. Selection of an improper fitness measure may result in;

- having good results for the training set but poor performance on the testing set,
- causing the iterative approaches to take long time to derive the model due to the selection of a computationally expensive fitness measure,
- being unable to derive the model by linear programming or least square based approaches due to an ill-defined nature of the chosen fitness measure [6].

To maximize the performance of the derived model, the fitness measure has to be chosen based on;

- the nature of the data set, and

- the experts' opinion; including the optimization engineer and the client who needs the model.

Conventionally some well-known fitness measures regardless of the nature of the problem are always utilized for optimization methods, or in more flexible approaches the selected fitness measures are tweaked such that they are converted to a convex optimization, or can be modeled by linear programming. Global continuous optimization approaches remove the restriction of using any type of a fitness function and allow to choose a fitness measure based on the client request or requirement, and the nature of the data set [6]. Moreover, for the applications where the fitness measure is not transparent to the designer of the fuzzy linear regression analysis due to the confidentially and secret trade, the model cannot be derived based on linear programming or least square based approaches [30]. This is due to the fact that in the both types of approaches, the fitness measure should be transparent to develop the model. In these scenarios where the fitness measure has to be treated as a black box, global continuous optimization approaches such as those which work based on meta-heuristics or surrogate methods are the only suitable solutions. Furthermore, one of the promises of the global continuous optimization approaches is to find the minimum of the fitness measure in minimum of time. This is due to the fact that the evaluation of the fitness measure is assumed to be a very time consuming process [2, 24, 30, 62]. So, in situations where the fitness measure is complex and time consuming to be evaluated, global continuous optimization approaches are suitable tools.

In a nutshell, the application of global continuous optimization for fuzzy linear regression allows;

- to choose the most suitable fitness measure based on the clients' request and/or requirement, and the nature of the data set,
- to keep the confidentiality of the fitness measure when it is a trade secret, and
- to have a fairly complicated and computationally expensive fitness measure––as one of aims of global continuous optimization is to find the minimum of the measure in a very short time.

3.3.2 Outlier detection

Cleaning the data set off outliers is an important preprocessing task. There are some methods in the literature which discuss the elimination of outliers before the application of fuzzy linear regression [7, 83–85]. A feasible way of detecting outliers is to develop a fitness measure such that it detects the outliers. This requires high flexibility on designing the fitness measure which is not always possible in least square or linear programming based approaches. Global continuous optimization approaches provide the flexibility of being able to apply any type of fitness measures. Chapter 6 proposes an outlier detection approach based on the application of global continuous optimization.

3.3.3 The spread increasing problem

Spread increasing problem has been recognized since the early stages of fuzzy linear regression. This is due to the fact that the fuzzy linear model which is initially proposed by Tanaka et al [5] and widely used by other researchers, naturally causes the uncertainty to increase with the increase in the magnitude of the input variable. The application of global continuous optimization allows to have a more flexible fuzzy linear model without being worried about the mathematical assumptions that one may encounter if we use least square or linear programming based approaches. In Chapter 6, we propose a fuzzy linear regression analysis method based on global continuous optimization which solves the spread increasing problem.

3.4 Summary

This chapter presents the preliminaries of fuzzy set theory for fuzzy linear regression analysis. Least square and linear programming based approaches are discussed as the main approaches to solve fuzzy linear regression. The motivations of applying global continuous approaches for non-transparent fitness measure due to confidentiality matters or mathematical assumptions are given. It is briefly described that how global continuous optimization provides solutions for non-transparency of the fitness measure, outlier detection and the spread increasing problem –the major issues in fuzzy linear regression analysis.

4

Approaching fuzzy linear regression by global continuous optimization[1]

We propose a meta-heuristic UGCO method based on tabu search and harmony search for FLR problem. The application of meta-heuristic approaches in UGCO allows us not to be concerned about the differentiability of the given function – a condition which must be satisfied when dealing with analytical approaches. So, any types of model fitting can be applied without being worried about how to model/solve problem or being concerned about differentiability or continuity of the chosen model fitting. The proposed method realizes the exploration of space by tabu search and then its further exploitation by harmony search. Tabu search is a robust technique to explore a wide area of the search space beyond the local optimality by positioning next solutions to unexplored areas [50]. It treats the objective function as a black box, which makes the optimization procedure context-independent. Harmony search is also a context-independent search procedure emulating some phenomena encountered in music. It employs a random search, which does not require the gradient of the function and hence makes it applicable to a wide range

[1] This chapter is based on a journal paper; [6] co-authored by M. A. Orgun, M. Mashinchi, and W. Pedrycz, and a conference paper; [89] co-authored by M. A. Orgun and M. Mashinchi

of optimization problems [127]. After finding a near-optimal solution, which is returned by tabu search, harmony search is then applied to exploit the surrounding area of such a near optimal solution to determine the global optimal solution.

The drawback of meta-heuristic approaches is their slow rate of convergence [24]. In other words, they can find the near minimal solution very quickly but it is a time consuming process to find a solution which is very close to the global minimum [2, 24]. That is the motivation to apply hybridized approaches which perform the diversification and then intensification in separate phases. Tabu search is applied for diversification and improved harmony search is used for intensification. Tabu search is chosen for two reasons. Firstly it is a point to point approach which makes it less computationally expensive in terms of the number of function evaluations compared to population based approaches such as genetic algorithms [2, 26]. Secondly, tabu search keeps track of the explored areas and stores them in a list so that it does not search the already explored areas. Thus it is less likely for tabu search to trap into local minima or become stuck in specific areas for a long time [50]. In the intensification phase an improved harmony search is applied. The advantage of harmony search over other meta-heuristic approaches like genetic algorithms is that in harmony search the improvision is mostly made for each vector individually rather than mating two vectors to generate offsprings. The mating approach is suitable when we intend to find the near optimal solution, which in our approach is already sought in the diversification phase by tabu search.

Fuzzy linear regression analysis can also be applied for extracting qualitative information from quantitative data when information is partially available [13], data are miss-recording due to inaccurate measurements [111].

4.1 Tabu-Harmony search optimization

The aim of UGCO methods is to search for the global minimum (maximum) of the given objective function. For a function of continuous variables X, we are interested in finding a vector of variables for which the function y attains its minimum. The optimality problem can be either concerned with finding the maximum or the minimum of the objective function. The mathematical formulation of UGCO for the minimization problem is given as follows:

$$
\begin{aligned}
y &= \min \; f(X) \\
f &: \mathbb{R}^n \to \mathbb{R}
\end{aligned}
$$

There are two different approaches in UGCO; analytical approaches and meta-heuristic ones. In the former approaches, the objective function should be known and has to satisfy conditions such as differentiability which is not necessary in meta-heuristic based approaches. In meta-heuristic based approaches, only the input-output behaviour of the function has to be provided.

4.1.1 The UGCO method

Tabu search is a sound technique for finding a near global solution as it can help from the local minima and search in a given search space. However, it is reported that it is not efficient to search for a solution that is located close to the global minimum [24]. This fact motivates us to hybridize tabu search with other local optimizer methods to increase the effectiveness of the overall search process. Harmony search is applied in the second phase after finding the near global optimal solution by tabu search. Geem et al. [127] proposed harmony search as a new meta-heuristic technique using the process of music generation. Similarly to tabu search, harmony search is a stochastic random search method which does not require the gradient of the function, thus making it easier to be applicable in a wide range of optimization problems without being concerned with the assumptions [127].

In this thesis, we hybridize tabu search and the improved harmony search as an UGCO method. The improvement in harmony search aims at making it more sutaible for local optimization. The near optimum solution obtained by the tabu search is provided to the improved harmony search. In the following subsections, the details of tabu search (the first phase of the UGCO method) and the improved harmony search (the second phase of the UGCO method) are discussed.

4.1.1.1 First Phase: Tabu Search

Among the meta-heuristic approaches, tabu search places a particular emphasis on escaping from traps of local minima. To accomplish this aim, tabu search keeps track of the last n visited solutions by placing it on a tabu list. Tabu search starts from a random solution positioned in the search space and then generates some neighbors around the current solution. Then it checks the function's values for the neighbors. The next solution can even have worse perfomance compared to the current solution and yet it can be still selected. This allows tabu search to explore a wider search area. But if the solution has been listed tabu, it can still be accepted as the next solution provided that it is better than the best solution reported so far. The algorithm continues until it satisfies a certain stopping condition [50]. The stopping condition can be set to a maximum number of iterations. Alternatively, the algorithm can stop if the distance between the solutions in the k^{th} and the $(k+n)^{th}$ iterations (where $k, n \epsilon \mathbb{N}$) is smaller than a small positive control value such as ϵ.

The starting solution always plays an important role in the rest of the optimization process. Starting from an initial point located far from the solution makes the optimization process very time consuming or even potentially trapped in a local minimum. To increase the speed of convergence and enhance the success rate, the initial solutions should be generated randomly to some extent but in a more systematic manner. Generating finite random solutions in a high-dimensional search space cannot normally cover the whole space. In virtue of that, we apply a partitioning process in which the search space is divided into partitions (sectors) and then solutions are generated in each of the partitions in a random manner. This allows the solutions to be distributed all around the search space while they are generated randomly in each partition. Assuming the domain of independent variables is $[a_i, b_i]$ (where i is the index of independent variables), we can divide each variable's domain into $m \epsilon \mathbb{N}$ equal intervals and then m^i random solutions are required to capture the partitions. The best solution is then selected as the current solution.

A move from the current solution to the next solution is carried out by a neighbor search strategy, which works based on the concept of randomization. In contrast to discrete tabu search, continuous tabu search has a more complicated neighbor search strategy as the number of neighbors around the current solution could be infinite. The basic mechanism is to generate random neighbors around the current solution with the maximum distribution radious of distribution factor (DF) and then the best one is selected according to its fitness. To distribute the generated neighbors well, DF is divided into crowns and one neighbor is generated for each of the crowns [24]. Thus the neighbors are distributed homogeneously around the current solution. The crown neighbor search strategy enables the neighbors to deliver a better approximation of the entire neighborhood space.

In continuous tabu search, a solution is considered as tabu if it is in the neighbourhood of a solution which has been seen before. The similarity factor (SF) defines the neighbourhood radius of each solution. If the SF is set as to a small value then tabu search works as a local optimizer. On the other hand, if SF is set to a large enough value, it behaves as a global optimizer. The process of generating random neighbors in the crowns with DF and tabulating the neighboorhod space of unseen solution with SF is shown in Figure 4.1.

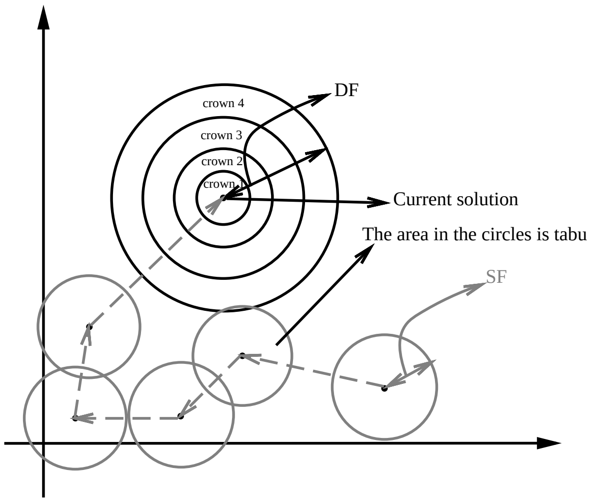


Figure 4.1: Generaing neighbors and tabulating areas based on DF and SF (taken from [6])

we consider two tabu lists: one for the short term and another one for the long term. A finite number of solutions recently visited are stored in the short term tabu list, so in the near future we avoid searching them again. In each iteration, if a local optimal solution is seen, then we save it into the long term tabu list. A solution is considered as a local minimum solution if the next solution exhibits worse performance. Simply speaking, if the current solution is still improving then we are getting close to the local optimum solution, otherwise the current solution should be a local optimum one if the next one has worse performance. The long term tabu list keeps track of local optimum solutions so later, in semi-intensification phase, the most potential one is searched.

The semi-intensfication is carried on the stored solutions in the long term tabu search to find out which of the local optima have the most potential to return a global optimum solution. The difference between semi-intensification and diversification is in applying smaller values for SF and DF. At the end of semi-intensification, only one point is nominated as the potential point for further intensification, which is followed by the improved harmony search.

4.1.1.2 Second Phase: Improved harmony Search

Harmony search is a meta-heuristic approach, which adopts the idea of natural musical processes [127]. The algorithm of harmony search minimizes an objective function of the form $f : \mathbb{R}^n \rightarrow \mathbb{R}$. In basic harmony search, randomly generated feasible solutions are initialized in the Harmony Memory (HM). In each iteration, the algorithm aims at improvising the HM. The improvision process works based on three operations; memory consideration, pitch adjustment and random selection. In the memory consideration and the random selection operations, each variable of a new solution vector is generated either based on the solutions in HM or randomly. The pitch adjustment operation, introduced to escape from local minima, makes random changes to some of the generated solutions [57,127].

The algorithm of the basic harmony search can be outlined as follows [127]:

1. Initialization of control variables.

2. Harmony Memory (HM) is initialized with HMS (Harmony Memory Size) randomly generated feasible solution vectors X^i where $i = 1, 2, \cdots, HMS$ from the solution $X_t = (x_t^1, x_t^2, \cdots, x_t^{n-1}, x_t^n)$ obtained from the tabu search. The initial HM is made up of the solution sought from tabu search plus $HMS - 1$ solutions which are chosen randomly in the neighborhood of X_t as follows:

```
**for** $i = 1$ to $HMS - 1$ **do**
    **for** $j = 1$ to $n$ **do**
        $x_j^i \Leftarrow x_t^i + RND(-0.5, 0.5)$;
    **end for**
**end for**
```

Then the solutions are sorted according to the output which is produced by the objective function, as shown below:

$$HM = \begin{bmatrix} x_1^1 & \dots & x_j^1 & \dots & x_n^1 \\ x_1^2 & \dots & x_j^2 & \dots & x_n^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^i & \dots & x_j^i & \dots & x_n^i \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{HMS-1} & \dots & x_j^{HMS-1} & \dots & x_n^{HMS-1} \\ x_1^{HMS} & \dots & x_j^{HMS} & \dots & x_n^{HMS} \end{bmatrix}$$

where $X_j^i = (x_1^i, x_2^i, \cdots, x_j^i, \cdots, x_{(n-1)}^i, x_n^i)$ and n is the number of variables to be optimized.

3. A new HM is improvised from the current HM. So, for each solution vector X^i, $\acute{X}^i$ represents the new solution vector which is going to be substituted for X^i. The next solution can be either chosen from the HM with the probability of the harmony memory consideration rate ($HMCR$), or generated randomly with the probability of $1 - HMCR$ in the feasible search space M. In this thesis $HMCR$ is set to be 0.9 as it is an intensification phase and we do not need too much mutation. This solution is then adjusted with a probability of pitch adjustment rate PAR and with the probability of $1 - PAR$ the solution remains intact. In pitch adjustment, the solution changes slightly in the neighborhood space of the solution. To be able to search the neighborhood of each solution comprehensively, we assign a large value like 0.8 to PAR. The entire process of improvising harmony memory can be summarized as follows:

Harmony memory consideration:

$$\overset{'}{\mathrm{X}}^{i} = \begin{cases} X^{i} \epsilon HM \\ \quad \text{with probability of } HMCR \\ X^{i} \epsilon M \\ \quad \text{with probability of } (1 - HMCR) \end{cases}$$

Pitch adjustment:

$$\overset{'}{\mathrm{X}}^{i} = \begin{cases} \overset{'}{\mathrm{X}}^{i} = \overset{'}{\mathrm{X}}^{i} + (bw \cdot RND(-1,1)) \\ \quad \text{with probability of } PAR \\ X^{i} \\ \quad \text{with probability of } (1 - PAR) \end{cases}$$

where bw is an arbitrary positive value for pitch adjustment which is usually assigned a value less than 1. In this thesis, bw is set to 0.1. The function $RND(-1,1)$ generates a vector of random numbers coming from the uniform distribution over $[-1,1]$. The role of $bw \cdot RND(-1,1)$ is to produce both negative and positive values randomly. So if we assign an arbitrary value in the closed interval of $[-1,1]$ to bw instead of $[0,1]$, then we still need to produce random values in the interval of $[-1,1]$.

4. The new solution vector $\overset{'}{X}^{i}$ is substituted with the worst solution in the HM – provided it outperforms the worst one.

5. If the stopping criterion is not met then GOTO 3.

As the harmony search is applied in the intensification phase, we alter the improvision part of the basic harmony search and the one proposed in global-best harmony search [128]. First, to increase the processing speed, we do not consider the process of pitch adjustment for all solutions, which are made after the harmony memory consideration phase. Thus if a solution is selected randomly with the probablity of $1 - HMCR$, then it does not need to undergo the pitch adjustment process [128]. Secondly, the pitch adjustment operation proposed by Omran and Mahdavi [128] based on the idea of particle swarm optimization becomes modified. In their approach, instead of making a random change in the generated solution after the harmony memory consideration phase, the solution is replaced with the best solution in HM with the probability of PAR. We extend this approach by giving a chance not only to the best solution but to all the solutions. Here in addition to extending the global-best concept in [128], we have pitch adjustment as well. So, in the proposed approach, if the pitch adjustment probability (PAR) is satisfied then the solution is replaced with one of the solutions in HM with the probability of an arbitrary control value *best* . The selection process of one of the solutions is based on the fitness of the solutions in the HM. According to their goodness, a probability of selection is assigned to each of the solutions. Thus the goodness probability (GP) of a solution like X^i with a goodness value of $f(X^i)$ is computed as follows:

$$GP^i = \frac{f(X^i)}{\sum_{i=1}^{HMS} f(X^i)}.$$

Then with the probability of $1 - best$, the solution is randomly adjusted/tuned. As this is the intensification phase, we set $best = 0.8$, so that we have less random adjustment and more focus on local optimization. The improvision step of the proposed method is given in the form of Algorithm 4.1.

Algorithm 4.1 Proposed algorithm of the improvision step

```
for $i = 1$ to $HMS$ do
   for $j = 1$ to $n$ do
      if $RND(0,1) \leq HMCR$ then
         $\overset{'}{\mathrm{X}}^{i}_{j} \Leftarrow X^i_j$;
         if $RND(0,1) \leq PAR$ then
            if $RND(0,1) \leq BEST$ then
               Based on the Goodness Probability (GP) of solutions, select one of
               the solutions and store the index of this solution into $b$;
               $\overset{'}{\mathrm{X}}^{i}_{j} \Leftarrow X^i_b$;
            else
               $\overset{'}{\mathrm{X}}^{i}_{j} \Leftarrow \overset{'}{\mathrm{X}}^{i}_{j} + (bw \cdot RND(-1,1))$;
            end if
         end if
      else
         randomly select $\overset{'}{\mathrm{X}}^{i}_{j} \in M$;
      end if
   end for
end for
```

4.2 Modeling fuzzy linear regression analysis as a global continuous optimization problem

we show how to model the FLR as an optimization problem so that it can be solved by an UGCO. There are some studies in the field of FR by means of meta-heuristic approaches. As an example we can refer to; genetic algorithms [110, 129], genetic programming [105, 130], tabu search [89], and fuzzy neural networks [131–133]. In this research, FLR is addressed by the proposed UGCO method based on tabu search and improved harmony search. For the applications of tabu search in fuzzy optimization problems, we refer the reader to the literature [50].

Finding the optimized FLR model over the given numeric input-fuzzy output data set $\left((x_{1i}, x_{2i}, \cdots, x_{(n-1)i}, x_{ni}), \widetilde{y}_i^*\right)$ [5] for $i = 1, \cdots, k$ with a minimum error can be viewed as a simple inputs-output system. This system receives the input vector and then produces the output based on the underlying function. The more general case of fuzzy input-fuzzy output [14] can be treated in the same way. Here, the underlying function should be the objective function (similarity measure) which shows the goodness of the FLR model. Thus the input vector consists of the coefficients of the FLR model and the output is the generated error of this FLR model over the given data, as shown in Figure 4.2.

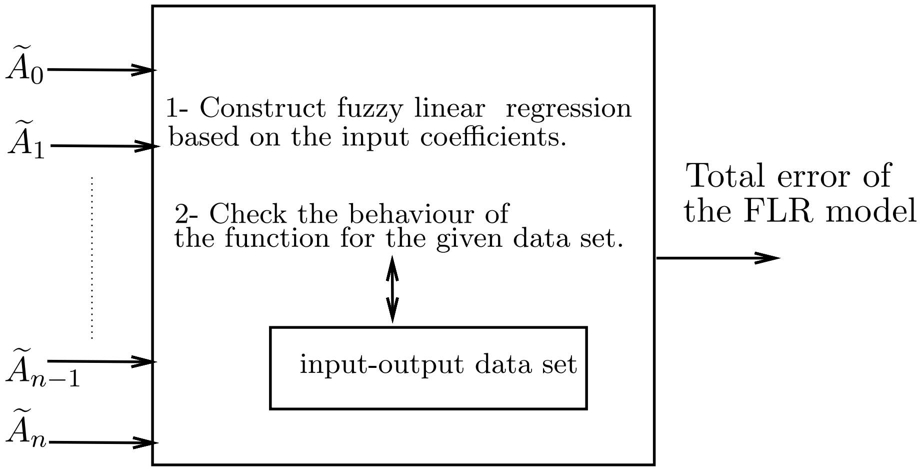

$\widetilde{A}_0$
$\widetilde{A}_1$
$\widetilde{A}_{n-1}$
$\widetilde{A}_n$
1- Construct fuzzy linear regression based on the input coefficients.
2- Check the behaviour of the function for the given data set.
input-output data set
Total error of the FLR model

Figure 4.2: Modelling of a FLR problem regarded as an UGCO problem (taken from [6])

Having the system presented in Figure 4.2, the UGCO method starts with a random initial input vector which is basically the initial FLR model. After a number of iterations of guessing new input vectors, the UGCO method tends to find a near global optimum solution of the function. The input vector which enables the system to obtain its global minimum contains the optimized coefficients of the FLR model which fits the given data to the highest extent.

To show the fitness (performance) of the FLR model, we compare the fuzzy output of the model $\widetilde{y}_i$ with the observed one $\widetilde{y}_i^*$, where i is the index of the given data (point). Minimizing the total difference between the observed fuzzy output and the estimated one is the goal of the optimization. There are different measures to determine the similarity between two fuzzy numbers. In this thesis, we use one of the most used objective functions; the *relative none-intersected area* (*RNIA*) which computes the difference area of observed and estimated outputs, as given in (4.1) [1,12,123,134,135]. The numerator of the *RNIA*, the none-intersected area, for two fuzzy numbers is illustrated in Figure (3.2). However, as mentioned earlier, the selection of *RNIA* is just for comparison purposes, and any other type of objective functions can be designed and then applied.

$$RNIA = \sum_{i=1}^{k} \left[\frac{\int_{S_{\widetilde{y}_i^*} \bigcup S_{\widetilde{y}_i}} |\mu_{\widetilde{y}_i}(x) - \mu_{\widetilde{y}_i^*}(x)| dx}{\int_{S_{\widetilde{y}_i^*}} \mu_{\widetilde{y}_i^*}(x) dx} \right]. \tag{4.1}$$

Here $S_{\widetilde{y}_i^*}$ and $S_{\widetilde{y}_i}$ are the supports of the observed output, $\widetilde{y}_i^*$, and the estimated output, $\widetilde{y}_i$, respectively. Note that if we consider the objective function in (4.1) alone, the possibility of being trapped into local minima is high. This happens as the method can be trapped at the zero input vector (one of the potential local minima), where the error is equal to 1. The other drawback of *RNIA* is that if $\widetilde{y}^*$ and $\widetilde{y}$ do not exhibit any overlap, the estimated similarity becomes constant regardless of their relative position [123]. To avoid these mentioned drawbacks, we combine *RNIA* with the level-set difference measurement (*LSM*) given in [110,116,134]. To describe the *LSM* procedure, we first start with preliminaries and relevant notation.

We can break down the fuzzy coefficients into some level-sets. In this way the input vector will have $(n+1) \times 2 \times \beta$ variables, where $n+1$ is the number of models coefficients and $2 \times \beta$ is the number of left and right spreads of β level-sets. For a fuzzy number like $\widetilde{A}$, where $\widetilde{A} = \{(x, \mu_{\widetilde{A}}(x))| \; x\epsilon\mathbb{R}, \; \mu_{\widetilde{A}} \rightarrow [0,1]\}$, the level-set α (or α-cut) of $\widetilde{A}$ is as follows.

$${}_{\alpha}A = \{x\epsilon\mathbb{R}| \;\; \mu_{\widetilde{A}}(x) \geq \alpha\}, \; \alpha\epsilon(0,1]. \tag{4.2}$$

In (4.2), ${}_{\alpha}A$ can be represented as $[{}_{\alpha}a^L, {}_{\alpha}a^R]$, where ${}_{\alpha}a^L$ and ${}_{\alpha}a^R$ are the left and right spreads of $\widetilde{A}$ at level set α.

Now we define the *LSM* measurement as follows.

$$LSM = \sum_{i=1}^{k}\left(\int_0^1 |{}_\alpha y_i^L - {}_\alpha y_i^{*L}| + |{}_\alpha y_i^R - {}_\alpha y_i^{*R}|\right) d\alpha \tag{4.3}$$

Here, ${}_\alpha y_i^L$, ${}_\alpha y_i^{*L}$ and ${}_\alpha y_i^R$, ${}_\alpha y_i^{*R}$ are the left and right spreads of the estimated and observed fuzzy output at level set α, respectively. Note that in (4.3), we assume ${}_0 y_i^L$, ${}_0 y_i^{*L}$, ${}_0 y_i^R$, ${}_0 y_i^{*R}$ are equal to zero.

For a normalized fuzzy number with a triangular membership function β is set to 2 as it can be represented by α-cuts of 0 and 1. We represent a triangular fuzzy number like $\widetilde{A}$ with the tuple (a^L, a^C, a^R), where a^C is the center of the membership function and a^L, a^R are the left and the right spreads of the membership function. In this case *LSM* given in (4.3) is simplified as follows:

$$LSM = \sum_{i=1}^{k} |y_i^L - y_i^{*L}| + |y_i^C - y_i^{*C}| + |y_i^R - y_i^{*R}| \tag{4.4}$$

The extended objective function (*EOF*) which combines both *RNIA* and *LSM* to decrease the possibility of being trapped into local minima is given in the following form:

$$\begin{aligned} EOF(\widetilde{A}_0, \widetilde{A}_1, \cdots, \widetilde{A}_n) = &\sum_{i=1}^{k}\left[\int_0^1 \left(|{}_\alpha \widetilde{y}_i^L - {}_\alpha \widetilde{y}_i^{*L}| + |{}_\alpha \widetilde{y}_i^R - {}_\alpha \widetilde{y}_i^{*R}|\right) d\alpha\right] \\ &+ \sum_{i=1}^{k}\left[\int_{S_{\widetilde{y}_i^*} \cup S_{\widetilde{y}_i}} |\mu_{\widetilde{y}_i(x)} - \mu_{\widetilde{y}_i^*(x)}| dx\right] \end{aligned} \tag{4.5}$$

The model fitting measurement given is (4.5) works only for numeric input-fuzzy output or fuzzy input-fuzzy output. However, other model fitting measurements can be applied to derive a linear regression model for numeric inputs and outputs.

To come up with a sound approximation for the search space (domain) of each coefficient, we apply statistical linear regression as a pre-processing phase. The given data are defuzzified (decoded to numeric values), and then the best linear function is sought to fit them. In defuzzification, we simply take the center point of the core for each fuzzy number. The neighborhood of the coefficients of the linear regression are later taken as the search domain for each fuzzy coefficient in FLR.

The objective function to find the optimized triangular normalized fuzzy numbers as the coefficients of the FLR model is defined as follows:

$$\begin{aligned}
y =& \min \ EOF(\widetilde{A}_0, \widetilde{A}_1, \cdots, \widetilde{A}_n)\\
&= \min \ EOF(a_0^R, a_0^C, a_0^L, a_1^R, a_1^C, a_1^L \cdots, a_n^R, a_n^C, a_n^L)\\
EOF :& [(k_1-\theta)-\gamma, (k_1-\theta)] \times [k_1-\theta, k_1+\theta] \times\\
&[k_1+\theta, (k_1+\theta)+\gamma] \times \cdots \times [(k_n-\theta)-\gamma, (k_n-\theta)] \times\\
&[k_n-\theta, k_n+\theta] \times [k_n+\theta, (k_n+\theta)+\gamma] \rightarrow \mathbb{R}
\end{aligned} \tag{4.6}$$

where k_0, k_1, $\cdots$, k_n are the coefficients of the linear function which fits the defuzzified data and θ and γ are neighborhood constants that define the domain of the fuzzy coeficients of FLR.

4.3 Method's configurations

We apply the UGCO method to the development of the FLR model. The data sets which are taken from the literature are divided into two main classes: numeric inputs-fuzzy output data and fuzzy inputs-fuzzy output data. These two classes just involve symmetric fuzzy numbers.

Since the proposed method is not just restricted to symmetric fuzzy numbers, we have validated our method in terms of asymmetric fuzzy input-asymmetric fuzzy output as the third class. Moreover, the performance of the model is investigated for a large data set with outliers.

Different models in the literature applied different model fitting measurements (MFMs) such as Hojati's similarity measure [4], distance criterion [99], Euclidean distance [100], none-intersected area [88, 101], relative none-intersected area [12], and compatibility measure [102]. Since there is no evidence that one measure is better than the others, one may adopt a MFM based upon requirements of modelling and the nature of the data set. Thus, in fairness to other studies in the literature, three other reported MFMs are applied for comparative analysis of models' performance. The first one is a similarity measure proposed by Hojati [4] given by equation (6.3), the second one is a distance criterion which is proposed in [99] expressed as equation (4.8), and the third one is the none-intersected area (NIA) proposed in [88, 101] and defined by equation (4.9).

$$S_h = \frac{\int \min(\mu_{\widetilde{y}^*(x)}, \mu_{\widetilde{y}(x)})dx}{\int \max(\mu_{\widetilde{y}^*(x)}, \mu_{\widetilde{y}(x)})dx} \tag{4.7}$$

$$D = \frac{1}{4}\sum_{i=1}^{k} |y_i^L - y_i^{*L}| + 2|y_i^C - y_i^{*C}| + |y_i^R - y_i^{*R}| \tag{4.8}$$

$$NIA = \sum_{i=1}^{k}\left[\int_{S_{\widetilde{y}_i^*} \bigcup S_{\widetilde{y}_i}} |\mu_{\widetilde{y}_i(x)} - \mu_{\widetilde{y}_i^*(x)}|dx\right]. \tag{4.9}$$

.

Note that the Hojati's original MFM (6.3) is a similarity measure with values between 0 and 1, so, the closer the value to 1, the better the model. However, for the sake of conformity with other MFMs, we have reported $1 - S_h$ which can be seen as a dissimilarity measure and the closer the value to 0, the better the model.

Table 4.1: Parameter settings for both phases of the proposed method

Diversification phase with TS

Parameter	*Value*
Cores neighborhood space (θ)	4
Spreads neighborhood space (γ)	2
Initial population size (IPS)	64
Population size	5
Number of iterations with no improvement (NI-first)	5
Tabu list size	10

Intensification phase with Improved HS

Parameter	*Value*
Population size	10
Neighborhood space (NS-second)	±0.5
Depth (Dep)	8
Number of iterations with no improvement (NI-second)	5
Decreament coefficient of neighborhood space (Dec)	2

For comparison of the models for a given data set, we say model f_1 is superior to model f_2 over $MFM_1, MFM_2, \cdots, MFM_n$, where all MFMs are minimization problems, only if all the following *"fuzzy regression model comparison conditions"* simultaneously hold:

$$\begin{aligned} MFM_1(f_1) &\leq MFM_1(f_2) \\ MFM_2(f_1) &\leq MFM_2(f_2) \\ &. \\ &. \\ &. \\ MFM_n(f_1) &\leq MFM_n(f_2) \end{aligned} \tag{4.10}$$

In case for a data for which set only some of the above conditions hold, it can be said that f_1 and f_2 are *"Pareto-equivalent"*. Note that in some cases f_1 can be superior to f_2 with respect to $MFM_1, MFM_2, \cdots, MFM_n$, however, if for a new MFM', f_2 is superior to f_1 then f_1 and f_2 are said to be Pareto-equivalent. So, for comparing the models for a data set, one should consider the client's requirements and requests and the nature of the data set for selecting suitable MFMs. However in this study, we restrict our comparisons based on RNIA, NIA, distance criterion and Hojati's measure as the applied data sets are mainly used for benchmarking purposes.

Four criteria given by (4.1), (4.9), distance criterion and Hojati's measure are reported for all the experiments. The similarity measure (4.9) is the total non-intersected area (NIA) between the observed and the estimated fuzzy outputs. Note that we have used (4.5) as the main criterion to carry out the optimization process, however the fuzzy regression model comparison conditions are applied to compare the models in terms of all the four measures. From now on, in the results, the values with and without rounded parentheses $(*)$ show the corresponding values of NIA and $RNIA$, respectively.

All meta-heuristic approaches require the setting of some control variables. The setting of the parameters is usually done by trial and error or based on some experts' intuitions. As an example in genetic algorithms one should set the mutation and crossover percentages, elite count, population size, selection method and so on. For the proposed method the parameters' settings for all the experiments are given in Table 4.1. To start, k_i in (4.6) which are the coefficients of the linear regression to fit the given defuzzified data are computed. For defuzzification, the mean of the inputs and outputs' core are considered. Then we generate IPS random fuzzy solutions around the coefficients. Intuitively, the fuzzy coefficients for FLR should be close to k_i, so we set our search space (domain) for FLR in (4.6) with small vales like

4 and 2 for "cores neighborhood constant" (θ) and "spread neighborhood constant" (γ), respectively . The initial population size plays an important role to arrive at a sound final result. By performing preliminary experiments, we noticed that any population size more than 50 produces good results, so, we set it to be between 60 to 70. A very large value for population size does not increase the performance considerably as much as they slow down the optimization process. The diversification phase runs until it does not find a better result for 5 consecutive iterations. We have observed if we set *NI-first* more than 5 the result will not improve significantly and it just makes the diversification phase very time consuming. Similarly, setting *NI-first* less than 5 increases the possibility of trapping into local minima.

In the intensification phase, we define a small search space as we already found the near optimum solution, so *NS-second* is set to 0.5. With the trail and error method, we found if the value of *NS-second* is greater than 1 then the intensification phase does more diversification and the use of values smaller than 0.1 makes the intensfication very time consuming. In contrast with the diversification phase, in the intensification phase the neighborhood space gets smaller if there is no improvement after *NI-second*. We set "Decrement Coefficient of Neighborhood Space" (*Dec*) to 2 as in the final stages the search space should become very small so the algorithm can find a solution very close to the optimum. The search space can become smaller up to "depth" times, so in the final stage of intensification the search space is as small as $\frac{NS-second}{Dec^{Dep}} = \frac{\pm 0.5}{2^8}$ which in here is equal to ±0.0019. As mentioned earlier the majority of the values shown in Table 4.1 are selected in an intuitive fashion and/or by the trial and error method.

As the proposed hybrid optimization method is non-deterministic, 1,000 separate runs with random seeds are attempted for each data set. The average, best and worst cases are then reported. In the literature of global continuous optimization with meta-heuristics, the speeds of the algorithms are compared based on their number of function evaluations rather than the execution time. This is due to the fact that the execution time of a method strictly depends on the computer's speed, the code optimization and the programming platform [2, 22]. So, for fair comparisons between the speeds of different global continuous optimization methods, only the number of function evaluations is considered. However, in this thesis, to show an approximation of how long the optimization process takes to compute a model, the execution time is given just for a couple of the datasets. The simulations were run on a PentiumIV 3.00GHz computer with 2GB of memory. The program is coded in Matlab 7.0. A detailed discussion of the experimental results are given in the next chapter.

4.4 Summary

Modeling and solving a FLR by the UGCO offers a number of benefits. When compared with the LS based approaches (which are the extension of classical regressions) and the probabilistic approaches (which aim at covering all the data), the proposed approach can use any type of model fitting. For example, one may design an objective function which combines LS and LP based model fitting measurements. The application of UGCO does not necessarily increase the spread of estimated fuzzy outputs. However, the spreads may increase if a smaller error is obtained. Moreover, unlike in fuzzy regression models such as those in [4,12,14,97,136], the proposed method is not just restricted to triangular fuzzy numbers and can be used for any form of fuzzy numbers (different membership functions). The method is capable of finding both positive and negative coefficients, while the methods presented in [4,14,88,97,101] are unable to process negative coefficients.

Fuzzy linear regression problem is approached by a global continuous optimization method. This method is the result of the hybridization of tabu search and improved harmony search. Tabu search is applied in the first phase and it searches the near optimum solution and then the improved harmony search intensifies the area close to the near optimum solution. The main benefit of applying this method over the existing methods is that no assumptions, such as differentiability or continuity, are required.

5

Performance evaluation of the proposed fuzzy linear regression analysis method[1]

To verify the performance of the proposed fuzzy linear regression method, we apply the method on crisp input-fuzzy output and fuzzy input-fuzzy output data sets from the literature. For comparing the results with other models in the literature, we use a comparison protocol which includes few fitting measures.

5.1 Numeric input-symmetric fuzzy output results

Example 1: Tanaka et al. in the first work on FLR defined the data set given in Table 5.1 to investigate the FLR model's performance. This data set has been used in many research studies as a benchmark, see [1, 4, 5, 12, 88, 97, 101, 109, 123, 135, 137]. This data set is an example of a small size numeric input-fuzzy output data. The results for 1,000 separate runs with random initial solutions are reported

[1] This chapter is based on a journal paper; [6] co-authored by M. A. Orgun, M. Mashinchi and W. Pedrycz and a conference paper; [89] co-authored by M. A. Orgun and M. Mashinchi

in Table 5.2. The generated errors produced by the estimated models for 1,000 runs are sorted and reported in Figure 5.1. The fuzzy regression model comparison conditions and the results given in equation (4.10) and Table 5.2 reveal that our model is Pareto-equivalent to the models given by Hojati et al. [4], Mashinchi et al. [89], Lu and Wang [86], and Hassanpour [90] and superior to the rest of the models over all of RNIA, NIA, distance criterion and Hojati's measure.

Table 5.1: Tanaka et al. data set [11]

Obs.	Independent variable	Dependent variable
1	1	$(6.2, 8.0, 9.8)$
2	2	$(4.2, 6.4, 8.6)$
3	3	$(6.9, 9.5, 12.1)$
4	4	$(10.9, 13.5, 16.1)$
5	5	$(10.6, 13.0, 15.4)$

Table 5.2: The experimental result for 1,000 runs for the data set given in Table 5.1 by the proposed method (taken from [6])

	best solution	worst solution	average	STD
Error (4.5)	11.7473	12.5346	12.0154	0.1760
Error (4.9)	8.3678	9.0414	8.5736	0.1426
Iterations	231	213	224	26
CPU time (s)	3.3326	3.0395	3.0811	0.2814

The models produced by the method presented here and other methods for the data set shown in Table 5.1 are given in Table 5.3. The proposed method has the smallest total error (4.9) and outperforms the other methods reported in the literature based on error given in (4.9). The plot of the estimated fuzzy function of some models for the given data in Table 5.1 is illustrated in Figure 5.2.

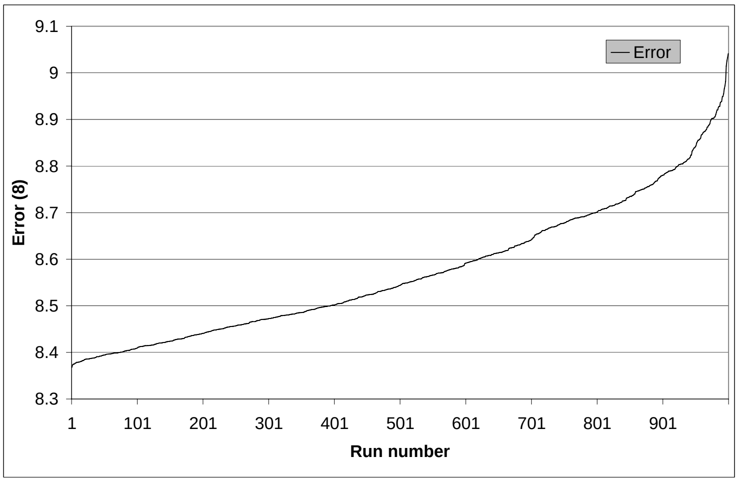


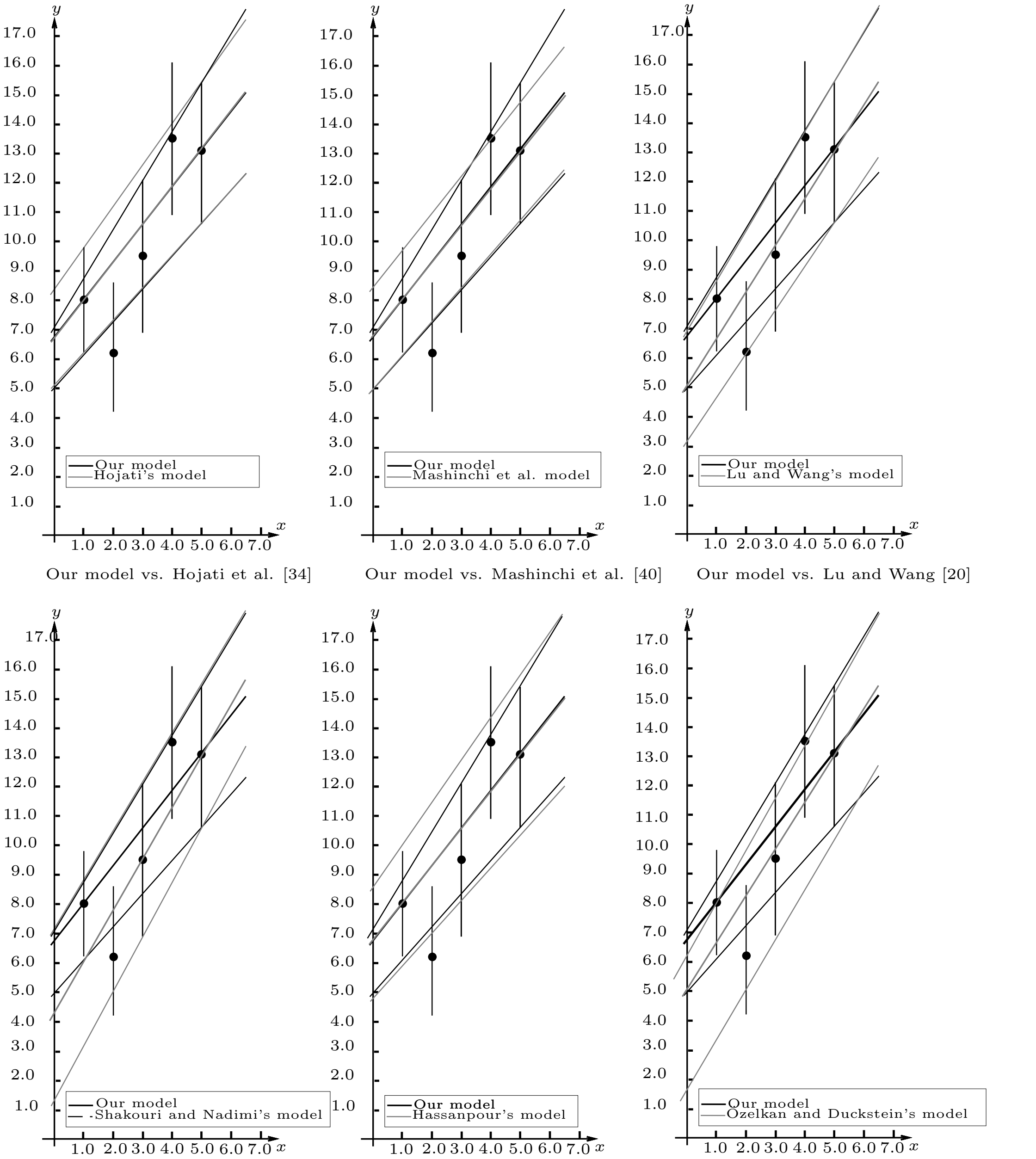
Figure 5.1: The sorted generated errors (4.9) for 1,000 separate runs for data set given in Table 5.1 obtained by the proposed method (taken from [6])

Table 5.3: Comparative analysis of fuzzy models in Table 5.1 (taken from [6])

Method	A_0	A_1
This work (best solution)	$(5.0798, 6.7652, 7.0613)$	$(1.1042, 1.2472, 1.6679)$
Chen and Hsueh (2009) [123]	$(2.63, 4.95, 7.27)$	$(1.71, 1.71, 1.71)$
Mashinchi et al. (2009) [89]	$(5.07, 6.72, 8.38)$	$(1.12, 1.27, 1.38)$
Lu and Wang (2009) [86][1]	$(4.25 + 1.75x_i, -0.10x_i + 2.90)$	
Shakouri and Nadimi (2009) [1]	$(3.2, 5.042, 6.884)$	$(1.480, 1.592, 1.704)$
Hassanpour (2008) [90]	$(4.95, 6.75, 8.55)$	$(1.05, 1.25, 1.45)$
Bargiela and Pedrycz (2007) [109]	$(4.95, 4.95, 4.95)$	$(1.719, 1.719, 1.719)$
Hojati, Bector and Smimou (2005) [4]	$(5.1, 6.75, 8.4)$	$(1.10, 1.25, 1.40)$
Modaress et al. (2005) [135]	$(2.98, 4.82, 6.66)$	$(1.50, 1.66, 1.82)$
Nasrabadi and Nasrabadi (2004) [136]	$(2.36, 4.68, 7.00)$	$(1.73, 1.73, 1.73)$
Kao and Chyu (2003) [101]	$(2.606, 4.926, 7.246)$	$(1.718, 1.718, 1.718)$
Kao and Chyu (2002) [88]	$(1.94, 4.95, 6.75)$	$(1.71, 1.71, 1.71)$
Özelkan and Duckstein (2000) [138][2] ($v = 25$)	$(3.4, 5.9, 8.4)$	$(1.4, 1.4, 1.4)$
Kim and Bishu (1998) [12]	$(3.11, 4.95, 6.84)$	$(1.55, 1.71, 1.82)$
Savic and Pedrycz (1991) [137][2]	$(0.92, 4.95, 8.98)$	$(1.64, 1.71, 1.78)$
Tanaka et al. (1989) [11][2]	$(0, 3.85, 7.7)$	$(2.1, 2.1, 2.1)$
Diamond (1988) [97][2]	$(3.11, 4.95, 6.79)$	$(1.55, 1.71, 1.87)$
Tanaka et al. (1982) [5]	$(0, 3.84, 7.69)$	$(2.1, 2.1, 2.1)$

[1] The estimated model for [86] has variable spread which we have shown it by $\widetilde{E_i}$.

[2] The estimated model for [97], [137], [11] and [138] are taken from [88], [12], [12] and [4], respectively.



Our model vs. Hojati et al. [34]

Our model vs. Mashinchi et al. [40]

Our model vs. Lu and Wang [20]

Our model vs. Shakouri and Nadimi [8]

Our model vs. Hassanpour [51]

Our model vs. Özelkan and Duckstein [53]

Figure 5.2: The estimated fuzzy function for the data given in Table 5.1: a comparative analysis involves six other models available in the literature (taken from [6])

Table 5.4: The average and generated error of the estimated models for the given data set in Table 5.1 (taken from [6])

Obs.	This work	Hojati et al. (2005) [4]	Mashinchi et al. (2009) [89]	Lu and Wang (2009) [86]	Shakouri and Nadimi (2009) [1]	Hassanpour (2008) [90]
1	0.5365 (0.2981)	0.0000 (0.0000)	0.03 (0.02)	3.13 (1.74)	2.2349 (1.24)	0.200 (0.1111)
2	3.3419 (1.5190)	3.7428 (1.7013)	3.71 (1.69)	2.33 (1.06)	2.8704 (1.30)	3.854 (1.7518)
3	1.5218 (0.5853)	1.7872 (0.6874)	1.78 (0.69)	0 (0)	0.6405 (0.27)	1.800 (0.6923)
4	2.9654 (1.1405)	2.8686 (1.1033)	2.86 (1.10)	3.51 (1.35)	3.2867 (1.26)	2.911 (1.1196)
5	0.0022 (9.16e-4)	0 (0)	0.12 (0.05)	0 (0)	0.0040 (0.00)	0.400 (0.1667)
Ave_e	1.6736 (0.7088)	1.6797 (0.6984)	1.70 (0.71)	1.79 (0.83)	1.8073 (0.8160)	1.8330 (0.7683)
NIA	8.3678	8.3986	8.51	8.96	9.0365	9.165
RNIA	3.5438	3.492	3.54	4.15	4.08	3.8415
Tot_e (Hojati's criterion)	0.494	0.4485	0.4642	0.4538	0.5186	0.4848
Tot_e (distance criterion)	5.6310	5.6	5.6825	5.6	5.6540	5.9

Continued...

Table 5.5: Continue from Table 5.4

Obj.	Özelkan and Duckstein (2000) [138]	Modarres et al. (2005) [135]	Kao and Chyu (2002) [88]	Nasrabadi and Nasrabadi (2004) [136]	Chen and Hsueh (2009) [123]
1	1.286 (0.714)	2.4320 (1.35)	2.7862 (1.5479)	2.564 (1.426)	2.324 (1.290)
2	3.474 (1.579)	2.7856 (1.27)	2.5901 (1.1773)	2.813 (1.277)	3.081 (1.400)
3	1.130 (0.434)	0.5817 (0.23)	0.5555 (0.2137)	0.718 (0.273)	1.092 (0.418)
4	3.216 (1.237)	3.2608 (1.25)	3.3604 (1.2925)	3.062 (1.179)	2.526 (0.976)
5	0.198 (0.082)	0.2971 (0.13)	0.3853 (0.1605)	0.614 (0.265)	0.947 (0.392)
Ave$_e$	1.860 (0.809)	1.8714 (0.8460)	1.9355 (0.8784)	1.954 (0.884)	1.996 (0.896)
NIA	9.303	9.3572	9.6776	9.771	9.976
RNIA	4.047	4.23	4.3919	4.421	4.477
Tot$_e$ (Hojati's criterion)	0.5048	0.5330	0.5337	0.5586	0.5886
Tot$_e$ (distance criterion)	5.7	5.78	6.1025	5.933	6.1
Continued...					

Table 5.6: Continue from Table 5.4

Obj.	Kim and Bishu (1998) [12]	Kao and Chyu (2003) [101]	Diamond (1988) [97]	Tanaka et al. (1989) [11]	Tanaka et al. (1982) [5]	Savic and Pedrycz (1991) [137]
1	2.207 (1.226)	2.217 (1.259)	2.2075 (1.2264)	3.364 (1.869)	3.3562 (1.8646)	2.7764 (1.54)
2	3.025 (1.375)	3.024 (1.396)	3.0499 (1.3863)	2.840 (1.291)	2.8500 (1.2955)	3.3308 (1.52)
3	1.041 (0.401)	1.082 (0.420)	1.0916 (0.4198)	1.512 (0.581)	1.5225 (0.5856)	1.7959 (0.7)
4	2.902 (1.116)	2.812 (1.083)	2.8444 (1.0940)	2.269 (0.873)	2.2578 (0.8684)	2.9968 (1.16)
5	0.850 (0.354)	0.954 (0.406)	0.9504 (0.3960)	2.404 (1.002)	2.4153 (1.0064)	2.0694 (0.86)
Ave$_e$	2.004 (0.894)	2.178 (0.913)	2.0288 (0.9045)	2.478 (1.123)	2.4803 (1.1241)	2.5939 (1.1560)
NIA	10.026	10.089	10.1438	12.388	12.4017	12.9693
RNIA	4.472	4.564	4.5225	5.616	5.6203	5.78
Tot$_e$ (Hojati's criterion)	0.5874	0.5905	0.5913	0.5687	0.5685	0.5631
Tot$_e$ (Distance criterion)	6.05	6.11	6.1	7.3	7.29	7.85

Example 2: The data set in this example given in Table 5.7 is related to cognitive response times of the nuclear power plant control room crew to an abnormal event and has been introduced by Kim and Bishu [12]. This is a benchmark data set for fuzzy multiple linear regression. The crew's fuzzy response time (in minutes) to an abnormal event is dependent on crews' experience inside a control room, experience outside a control room and training. This data set has been applied as a benchmark for comparison of the methods presented in [1,12,135]. The estimated model of this work in comparison with other works is given in Table 5.9. Based on (4.10) and the results of our comparative analysis shown in Table 5.10, our method is superior to the other models over RNIA, NIA, distance criterion and Hojati's measure.

Table 5.7: Kim and Bishu's data set [12]

Obs.	Dependent variable *(Response time)*	Independent variable *(Inside control room experience)*	Independent variable *(Outside control room experience)*	Independent variable *Education*
Team 1	$(5.83, 3.56)$	2.0	0.0	15.25
Team 2	$(0.85, 0.52)$	0.0	5.0	14.13
Team 3	$(13.93, 8.5)$	1.13	1.5	14.13
Team 4	$(4, 2.44)$	2.0	1.25	13.63
Team 5	$(1.65, 1.01)$	2.19	3.75	14.75
Team 6	$(1.58, 0.96)$	0.25	3.5	13.75
Team 7	$(8.18, 4.99)$	0.75	5.25	15.25
Team 8	$(1.85, 1.13)$	4.25	2.0	13.5

The results given in Table 5.8 show the error and the number of iterations for the case of the best and the worst solutions for 1,000 separate runs. As an example, Figure 5.3 illustrates the convergence of the error for the best run. It can be observed that the first phase is carried out for 125 iterations and then the intensification phase is started. In the first phase, the average error in each iteration exhibits some oscillations, which is due to the nature of tabu search applied in the diversification phase. The oscillations show the effectiveness of the tabu search which pushes the next solutions to an area which has not been searched so far and does not necessarily exhibit higher potential. By doing this, we can be almost sure that during the diversification phase we have effectively searched the whole search space. In the intensification phase, the average error in each iteration decreases smoothly with no oscillations as in each iteration, we get closer to the actual optimal solution.

Table 5.8: The experimental result for 1,000 runs for the data set given in Table 5.7 (taken from [6])

	best solution	worst solution	average	STD
Error (4.5)	29.7202	34.6411	12.0154	1.2071
Error (4.9)	21.9369	25.8395	23.7067	0.4984
Iterations	300	287	301	24
CPU time (s)	5.1921	5.1049	5.0826	0.327

Example 3: The data set in Table 5.11 was proposed by Tanaka et al. [5] and is concerned with the prices of housing in Japan. This gives rise to the fuzzy multiple regression analysis as the price variable is dependent on five numeric variables; the rank of material (x_1), the first floor space in square meters (x_2), the second floor space in square meters (x_3), the number of rooms (x_4), and the number of Japanese-style rooms (x_5), given in Table 5.11. This data set has the largest number of records in comparison with other data sets and is a good benchmark for fuzzy multiple linear regression and negative coefficients.

This data set has been applied to study the efficiency of the FLR methods presented in [5,87,113,139,140]. Table 5.11 shows the error produced by each method in comparison to the model which is given in (5.1). According to the error analysis reported in Table 5.11, the model presented here is Pareto-equivalent to the Kao and Lin's model [139] and superior to Tanaka et al.'s model [5].

$$\begin{aligned} &y = (2429.0, 2430.6, 2458.4)x_1 + (83.46, 83.48, 83.55)x_2 + \\ &(82.27, 84.69, 87.21)x_3 + (-402.54, -386.891, -383.892)x_4 + \\ &(-222.55, -220.64, -206.89)x_5 + (-1134.7, -1072.0, -1042.9) \end{aligned} \tag{5.1}$$

The performance of the method reported for 1,000 separate runs is given in Table 5.12. The convergence error for the best model is given in Figure 5.4. In the first 200 generations, the method is concerned with the diversification phase, and here we observe some oscillations due to searching in different parts of the search domain.

Table 5.9: Models available in the literature for the data set in Table 5.7 (taken from [6])

Method	Estimated function
This work	$(-16.44,-16.25,-16.09)+(-0.77,-0.31,-0.12)x_1+(-1.17,-1.14,-1.13)x_2+(1.60,1.61,1.62)x_3$
Shakouri and Nadimi (2009) [1]	$(-20.08,-20.08,-20.08)+(-0.23,-0.16,-0.09)x_1+(-1.22,-0.9,-0.58)x_2+(1.66,1.81,1.96)x_3$
Modarres et al. (2005) [135][1]	$(-23.0624-10.0507x_1-1.6288x_2+2.3474x_3,-10.2731-0.6696x_1-0.7199x_2+1.1341x_3)_L$
Kim and Bishu (1998) [12]	not reported
Savic and Pedrycz (1991) [137]	not reported
Tanaka et al. (1989) [11]	not reported

[1] This model is represented in the form of $(y_i^c, \alpha_{y_i})_L$ which is equal to $(y_i^c - \alpha_{y_i}, y_i^c, y_i^c + \alpha_{y_i})$.

Table 5.10: The average and generated error of the estimated models for the given data set in Table 5.7 (taken from [6])

	This work	Shakouri and Nadimi[4]	Kim and Bishu[3]	Modarres et al.	Tanaka et al.[3]	Savic and Pedrycz[3]
		(2009) [1]	(1998) [12]	(2005) [135]	(1989) [11]	(1991) [137]
Team 1	3.07 (0.86)	2.43 (0.68)	5.64 (1.58)	7.52 (2.11)	9.80 (2.75)	10.29 (2.89)
Team 2	0.10 (0.19)	3.20 (6.16)	2.66 (5.13)	2.20 (4.23)	8.87 (17.07)	9.86 (18.97)
Team 3	9.23 (1.09)	11.05 (1.30)	10.48 (1.23)	10.09 (1.18)	2.78 (0.32)	7.89 (0.92)
Team 4	1.49 (0.61)	1.56 (0.64)	1.53 (0.63)	2.10 (0.86)	10.46 (4.29)	8.19 (3.35)
Team 5	1.15 (1.14)	2.82 (2.79)	2.84 (2.81)	2.69 (2.66)	6.49 (6.42)	9.56 (9.47)
Team 6	0.56 (0.58)	2.24 (2.33)	3.37 (3.51)	2.92 (3.04)	10.57 (11.01)	10.02(10.44)
Team 7	5.69 (1.14)	7.65 (1.53)	5.62 (1.12)	6.30 (1.26)	4.90 (0.98)	5.83 (1.16)
Team 8	0.63 (0.56)	1.83 (1.62)	1.05 (0.93)	1.02 (0.90)	6.73 (5.95)	7.49 (6.63)
NIA	21.93	32.78[2]	33.25	34.85	60.64[1]	69.17[1]
Tot_e RNIA	6.18	17.06[2]	16.98	16.17	48.82[1]	53.87[1]
T_e (Hojati's criterion)	0.6521	0.7382	0.8217	0.84	0.7989	0.8017
T_e (Distance criterion)	20.3129	23.3340	23.6700	24.0550	42.9700	41.9500

[1] The estimated models error of Savic91 [137] and Tanaka89 [11] were reported in [12].

[2] According to the estimated function given in Table 5.9 which is taken from [1], the Shakouri's model error will be $NIA = 32.78$ and $RNIA = 17.06$. But the error was reported $RNIA = 5.16$ in Shakouri09 [1]-which is incorrect.

[3] The estimated outputs are taken from Modaress et al. [135] as the estimated models are not reported in the original papers.

[4] The reported fuzzy outputs and their corresponding errors from Shakouri [1] and the corrected ones are given in Table A.9 in the Appendix.

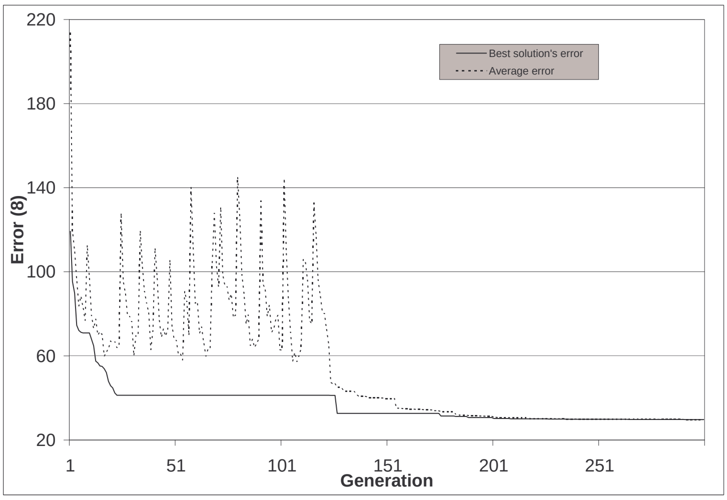


Figure 5.3: The convergence of the error for the best model over the data set in Table 5.7 (taken from [6])

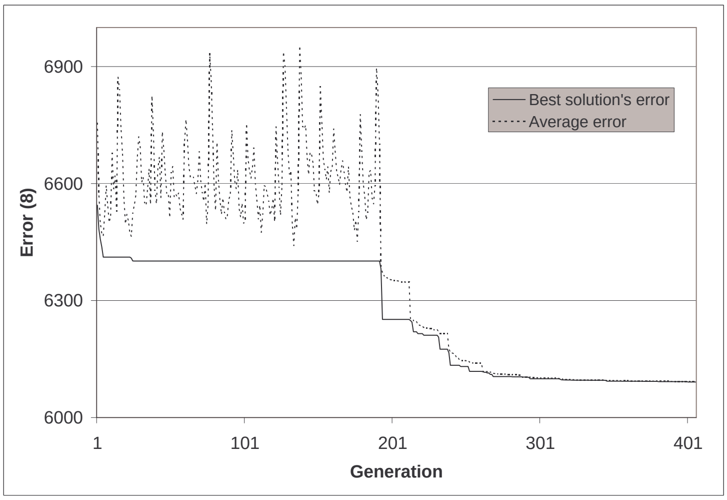


Figure 5.4: The convergence of the error for the best model over the data set in Table 5.11 (taken from [6])

TABLE 5.11: Tanaka et al. house price data set [5]

Obs.	Dependent variable	Independent variable	Model's error		
	$\widetilde{y}^*$	$(x_1, x_2, x_3, x_4, x_5)$	This work	Kao and Chyu (2005) [139]	Tanaka et al. (1982) [5]
1	$(6060, 550)$	$(1, 38.09, 36.43, 5, 1)$	390.27 (0.7096)	577.5 (1.05)	187 (0.34)
2	$(7100, 50)$	$(1, 62.10, 26.50, 6, 1)$	204.71 (4.0942)	431.5 (8.63)	663.5 (13.27)
3	$(8080, 400)$	$(1, 63.76, 44.71, 7, 1)$	252.56 (0.6314)	132 (0.33)	528 (1.32)
4	$(8260, 150)$	$(1, 74.52, 38.09, 8, 1)$	182.24 (1.2150)	307.5 (2.05)	591 (3.94)
5	$(8650, 750)$	$(1, 75.38, 41.40, 7, 2)$	541.30 (0.7217)	405 (0.54)	7.5 (0.01)
6	$(8520, 450)$	$(2, 52.99, 26.49, 4, 2)$	609.20 (1.3538)	558 (1.24)	625.5 (1.39)
7	$(9170, 700)$	$(2, 62.93, 26.49, 5, 2)$	664.66 (0.9495)	455 (0.65)	259 (0.37)
8	$(10310, 200)$	$(2, 72.04, 33.12, 6, 3)$	2.78 (0.0139)	252 (1.26)	912 (4.65)
9	$(10920, 600)$	$(2, 76.12, 43.06, 7, 2)$	606.42 (1.0107)	558 (0.93)	486 (0.81)
10	$(12030, 100)$	$(2, 90.26, 42.64, 7, 2)$	328.48 (3.2848)	500 (5.00)	947 (9.47)
11	$(13940, 350)$	$(3, 85.70, 31.33, 6, 3)$	328.81 (0.9395)	276.5 (0.79)	990.5 (2.83)
12	$(14200, 250)$	$(3, 95.27, 27.64, 6, 3)$	52.94 (0.2118)	295 (1.18)	820 (3.28)
13	$(16010, 300)$	$(3, 105.98, 27.64, 6, 3)$	505.30 (1.6844)	696 (2.32)	1104 (3.68)
14	$(16320, 500)$	$(3, 79.25, 66.81, 6, 3)$	251.65 (0.5033)	486 (0.81)	530 (1.06)
15	$(16990, 650)$	$(3, 120.5, 32, 25, 6, 3)$	524.19 (0.8065)	273 (0.42)	780 (1.20)
Average			361.06 (1.23)	413.53 (1.81)	616.26 (3.17)
NIA			5445.6	6203	9244
RNIA			18.13	27.20	47.62
T_e (Hojati's criterion)			0.7055	0.6763	0.7294
T_e (Distance criterion)			4.5835e+003	4.2524e+003	7.1557e+003

Table 5.12: The experimental result for 1,000 runs for the data set given in Table 5.11 (taken from [6])

	best solution	worst solution	average	STD
Error (4.5)	6355.18	8645.13	7098.08	354.67
Error (4.9)	5415.95	7539.95	6140.47	314.46
Iteration	406	410	426	25

Example 4: The data set given in Table 5.13 is proposed by Hong et al. [13]. What makes this data set different from the others is that negative fuzzy outputs are also considered. The proposed model and the comparison of the resulting error with other methods [13,103,140] are given in Table 5.14. The results show that our model is superior to the other models in the literature over NIA, RNIA, distance criterion and Hojati's measure. While there are many methods reported in the literature, which are not capable of finding functions with negative coefficients (e.g. [4, 14, 88, 97, 101]), our method is able to find a function with negative coefficients. Table 5.15 reports the performance of the proposed method for 1,000 separate runs. Even the worst produced error (NIA) by the method is smaller when compared with the errors produced by other methods presented in the literature.

Table 5.13: Hong and Hwang's data set [13]

Obs.	Independent variable	Dependent variable
1	1	$(-1.6, 0.5)$
2	3	$(-1.8, 0.5)$
3	4	$(-1.0, 0.5)$
4	5.6	$(1.2, 0.5)$
5	7.8	$(2.2, 1.0)$
6	10.2	$(6.8, 1.0)$
7	11.0	$(10.0, 1.0)$
8	11.5	$(10.0, 1.0)$
9	12.7	$(10.0, 1.0)$

Table 5.14: Estimated models and their errors for the data set given in Table 5.13 (taken from [6])

Method	A_0	A_1	NIA	RNIA	T_e (Hojati's criterion)	T_e (Distance
This work	$(-5.85, -5.83, -5.82)$	$(1.21, 1.24, 1.33)$	6.89	10.29	0.7056	9.14
Arabpour and Tata (2008) [103]	$(-5.25, -4.90, -4.54)$	$(1.14, 1.20, 1.25)$	10.3944	16.63	0.8446	9.98
Yao and Yu (2006) [140]	$(-7.88, -4.89, -3.29)$	$(1.20, 1.20, 1.120)$	19.7039	31.24	0.8027	13.29
Hong and Hwang (2003) [13]	$(-5.45, -5.45, -5.45)$	$(0.71, 1.22, 1.73)$	34.8130	42.42	0.8250	19.15

Table 5.15: The experimental result for 1,000 runs for the data set given in Table 5.13 by the proposed method (taken from [6])

	best solution	worst solution	average	STD
Error (4.5)	9.96	10.51	10.27	0.37
Error (4.9)	6.89	7.30	7.03	0.07
Iteration	229	244	210	24

5.2 Symmetric fuzzy input-fuzzy output results

This class involves a more generalized form of numeric input-fuzzy output. Sakawa and Yano [14] introduced a fuzzy input-fuzzy output data set, which is given in Table 5.16.

Many approaches present in the literature (in [4,14,86–88,97,101,109,141]) have used this data set for validation purposes. The result of the comparison between our estimated model and other methods' model is given Table 5.18. The comparison results show that the model generates a smaller error $RNIA$ than reported for other models generated by other techniques. However, based on the fuzzy regression model comparison conditions, our model is superior to Hassanpour's model [90] and Pareto-equivalent to the rest of the models over $RNIA$, NIA, distance criterion and Hojati's measure. Figure 5.5 illustrates the behaviour of the estimated fuzzy linear model for the data set given in Table 5.16. The dashed and continuous lines show the estimated fuzzy output and the observed outputs, respectively. According to the reported results in Table 5.17, the average error $RNIA$ for 1,000 separate runs is lower in comparison to the results produced by other models.

Table 5.16: Sakawa and Yano's data set [14]

Obs.	Independent variable	Dependent variable
1	(1.5, 2, 2.5)	(3.5, 4.0, 4.5)
2	(3.0, 3.5, 4.0)	(5.0, 5.5, 6.0)
3	(4.5, 5.5, 6.5)	(6.5, 7.5, 8.5)
4	(6.5, 7.0, 7.5)	(6.0, 6.5, 7.0)
5	(8.0, 8.5, 9.0)	(8.0, 8.5, 9.0)
6	(9.5, 10.5, 11.5)	(7.0, 8.0, 9.0)
7	(10.5, 11.0, 11.5)	(10.0, 10.5, 11.0)
8	(12.0, 12.5, 13.0)	(9.0, 9.5, 10.0)

Table 5.17: The experimental result for 1,000 runs for the data set given in Table 5.16 (taken from [6])

	best solution	worst solution	average	STD
Error (4.5)	7.6404	9.3234	7.8158	0.1913
Error (4.9)	5.2656	6.1336	5.4919	0.1582
Iterations	229	227	187	25

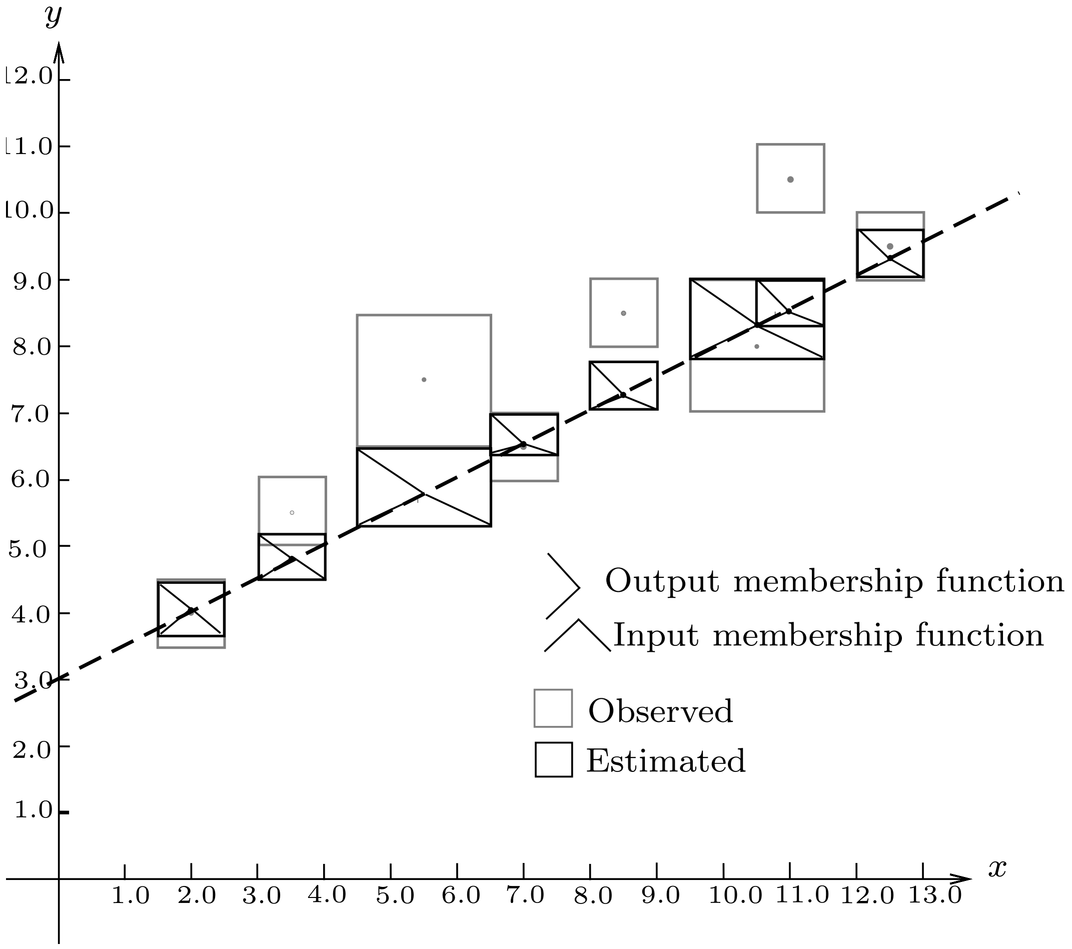


Figure 5.5: The estimated fuzzy function for the given data in Table 5.16 (taken from [6])

Table 5.18: Estimated models and generated errors for several models and data shown in Table 5.16 (taken from [6])

Method	A_0	A_1	NIA	RNIA	T_e (Hojati's criterion)	T_e (Distance) criterion)
This work	$(2.9606, 2.9679, 3.0042)$	$(0.5154, 0.5154, 0.5154)$	5.265	8.3046	0.7508	6.2460
Mashinchi et al. (2009) [89]	$(3.00, 3.02, 3.20)$	$(0.50, 0.50, 0.50)$	5.330	8.4608	0.7093	6.2050
Hassanpour GPA2 (2008) [90]	$(3.9444, 3.9444, 3.9444)$	$(0.4444, 0.4444, 0.4444)$	5.737	9.1423	0.8571	5.8340
Chen and Hsueh [99] (2007)	$(3.667, 3.945, 3.2230)$	$(0.444, 0.444, 0.444)$	6.246	9.979	0.6928	5.5640
Hassanpour GPA1 (2008) [90]	$(2.6154, 2.6154, 2.6154)$	$(0.6923, 0.6923, 0.6923)$	6.285	9.4093	0.7658	6.5769
Lu and Wang (2009) [86][3]	$(2.9524 + 0.5238x_i^c$ $, 2.3516\alpha_{x_i} - 0.6758))_L$		6.340	9.4920	0.5939	5.9452
Yeh (2009) [121]	$(3.530, 3.530, 3.530)$	$(0.5250, 0.5250, 0.5250)$	6.699	10.628	0.9169	5.8900
Arabpour and Tata method 1 (2008) [103][2]	$(3.4877, 3.4877, 3.4877)$	$(0.5306, 0.5306, 0.5306)$	6.739	10.687	0.9191	5.8968
Bargiela et al. (2007) [109]	$(3.4467, 3.4467, 3.4467)$	$(0.5360, 0.5360, 0.5360)$	6.776	10.742	0.9212	5.9033
Chen and Dang (2008) [87]	$(3.5284, 3.5284, 3.5284) + \widetilde{E}_i$	$(0.5298, 0.5298, 0.5298)$	7.000	10.808	0.8146	5.3868
Kao and Chyu (2002) [88]	$(3.3324, 3.5724, 3.8124)$	$(0.5193, 0.5193, 0.5193)$	7.470	12.030	0.8501	5.8175
Nasrabadi and Nasrabadi (2004) [136][4]	$(3.5767, 3.5767, 3.5767)$	$(-0.453, 0.5467, 1.5467)$	7.541	11.902	0.7948	5.9166
Chen and Hsueh [123] (2009)	$(3.2720, 3.5720, 3.8720)$	$(0.519, 0.519, 0.519)$	7.668	12.352	0.8375	5.8165
Arabpour and Tata method 2 (2008) [103]	$(3.2758, 3.5724, 3.8386)$	$(0.5188, 0.5193, 0.5235)$	7.691	12.382	0.8388	5.8208
Diamond (1988) [97][1]	$(3.2636, 3.5632, 3.8628)$	$(0.5206, 0.5206, 0.5206)$	7.709	12.379	0.8387	5.8221
Modarres et al. (RNP) (2004) [142][2]	$(3.278, 3.511, 3.744)$	$(0.542, 0.553, 0.562)$	7.783	12.448	0.8110	5.9337
Yang and Ku (2002) [141]	$(3.2052, 3.4967, 3.7882)$	$(0.5251, .5293, 0.5335)$	7.867	12.682	0.8429	5.8525
Hojati et al. (2005) [4][2]	$(3.00, 3.41, 3.82)$	$(0.50, 0.52, 0.54)$	8.284	13.464	0.7920	5.8200
Modarres et al. (RAP) (2004) [142][2]	$(3.152, 3.525, 3.898)$	$(0.485, 0.537, 0.589)$	9.169	14.995	0.7727	6.1200
Kao and Chyu (2003) [101]	$(2.603, 3.554, 4.503)$	$(0.522, 0.522, 0.522)$	9.363	15.966	0.8358	5.9575
Sakawa and Yano (1992) [14]	$(3.031, 3.201, 3.371)$	$(0.498, 0.579, 0.659)$	9.431	15.272	0.7717	6.1801

[1] The estimated models error of Diamond [97] is taken from [87, 88].

[2] The estimated models error of Arabpour and Tata [103], Modarres et al. [142] and Hojati et al. [4] are taken from [90].

[3] This model is represented in the form of $(y_i^c, \alpha_{y_i})_L$ which is equal to $(y_i^c - |\alpha_{y_i}|, y_i^c, y_i^c + |\alpha_{y_i}|)$.

[4] This model applied new addition and multiplication operators.

5.3 Numeric input-asymmetric fuzzy output results

To show the performance of the proposed method for numeric input-asymmetric fuzzy output, we have designed an artificial benchmark (Table 5.19). Although there is a numeric input-asymmetric fuzzy output in the literature for fuzzy multiple linear regression [100], the reason for introducing a new data set is to investigate the performance of the approach for a data set with very large differences in the spreads of each datum. If the difference in the spreads is not noticable then even the approaches which are only capable of having symmetric coefficients (such as [4]) may result in almost the same error level as the ones with asymmetric coefficients. After applying the method to this data set, the best model comes in the form (5.2). The total errors for the obtained model in (5.2) with NIA, Hojati's measurement, and distance criterion are 5.20, 0.3364, and 3.6701, respectively.

$$y = (0.1561, 0.1630, 0.1634)x_1 + (7.72, 8.33, 10.44) \tag{5.2}$$

As the method was run for 1,000 times the relevant statistics are reported in Table 5.20. The plot of the best model given by (5.2) is included in Figure 5.6. The experiments show that the method can deal with asymmetric fuzzy data effectively. It is quite expected that as five (out of six) points of the data set have larger right spreads than left spreads, the model follows this reflected tendency.

Table 5.19: Artificial asymmetric numeric input-fuzzy output (taken from [6])

Obs.	Independent variable	Dependent variable
1	1	(7.0, 8.0, 15.0)
2	2	(8.10, 8.50, 10.80)
3	3	(8.20, 10.0, 10.20)
4	4	(8.40, 9.40, 11.0)
5	5	(8.50, 9.20, 11.20)
6	6	(7.50, 8.30, 12.0)

5.4 Summary

The models derived from the proposed approach are tested against crisp input-fuzzy output, and fuzzy input-fuzzy output data sets available in the literature. In light of the experimental results, our method is either superior or Pareto-equivalent

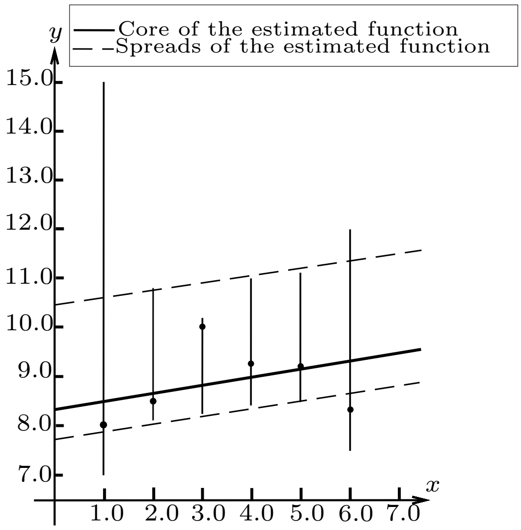


Figure 5.6: The estimated fuzzy function for the data set given in Table 5.19 (taken from [6])

Table 5.20: The experimental result for 1,000 runs for the data set given in Table 5.19 by the proposed method (taken from [6])

	best solution	worst solution	average	STD
Error (4.5)	7.10	9.02	7.60	0.32
Error (4.9)	5.20	6.45	5.54	0.21
Iterations	122	108	125	17

to the other methods discussed in the literature for all the given benchmark data sets. The method is tested against crisp input-fuzzy output, and fuzzy input-fuzzy output data sets available in the literature.

Moreover an approach based on fuzzy linear regression is applied to extract the qualitative information which may have been overlooked in the original data set. The extracted qualitative data is then used for modeling the data. The proposed approach is applied for the trust prediction of the delivered web-services based on a set of advertised QoS values. The method not only provides more detailed information of trust but it also has promising results. Although in order to quantitatively compare the performance of the method, the results are defuzzified, one could use the fuzzy results as they provide more information.

6

Approaching outlier detection and spread increasing problem by global continuous optimization[1]

Fuzzy linear analysis may lead to an incorrect interpretation of data in case of being incapable of dealing with outliers. Both basic probabilistic and least squares approaches are sensitive to outliers. In order to detect the outliers in data, we propose a two stage least squares approach which in contrast to the other proposed methods in the literature does not have any user defined variables. In the first stage of this approach, the outliers are detected and the clean dataset is prepared and then in the second stage a model is sought to fit the clean dataset. In both the first and second phases, the minimization of the model fitting measurement is achieved with hybrid optimization which gives us the flexibility of using any type of a model fitting measure regardless of being continuous or differentiable.

Fuzzy linear regression models which constantly increase the spreads of the outputs with the increase in the magnitude of the inputs, are known to have spread

[1] This chapter is based on a conference paper; [8] co-authored by M. A. Orgun and M. R. Mashinchi, and a sketch; [9]

increasing problem. We introduce two algebraic operations and then by application of global continuous optimization we overcome spread increasing problem. The models obtained by the proposed approach are capable of having variable spreads for different input variables regardless of their magnitute.

6.1 Outlier detection

Both probabilistic and least squares approaches are vulnerable to outliers unless there is a modification made in the method. Some modifications to probabilistic approaches are given in [7, 83–85]. Detecting outliers in fuzzy regression analysis is shown to be necessary in different real-world applications [143, 144]. Outliers happen due to miss collection, measurement or recording the data [83]. These are the common reasons of having outliers in crisp observations as well. However, in case of fuzzy data, modeling the subjectiveness of the observations may cause outliers. So basically we may have an expert who advises an almost perfect observation (which can be regarded as measuring), and then the system collects and records the measured observations with no problems, but at modelling stage, the fuzzy data is not modelled as it is measured/meant by the expert and consequently results in outliers.

As an example, let us assume that we are trying to find a fuzzy linear function of the temperature and the condition of the sea. Suppose that the sea condition can be represented in terms of being calm, tidy, or very tidy. Each of the experts live in different climates so they have experience in different ranges of temperature. For example, say, the first one is from Antactica so he knows the sea condition just for the temperature from -20 to 0. So, the temperature ranges are as $[a,b]$, $(b,c]$, $(c,d]$, $(d,e]$ and $(e,f]$. From each expert, the sea condition is questioned for two different temperature in their own temperature range and their opinion is collected and stored perfectly. However the subjectiveness of data which is hidden in linguistic expressions of calm, tidy and very tidy can be interpreted in different ways. The expert who is in a very cold place has a different feeling for the meaning of calm from that of someone in a very hot place. This causes the spreads or even the core of the fuzzy observation to become outliers if the person who converts the expressions into numbers does not really understand the subjectiveness of each expert.

In the literature, different probablistic approaches are proposed to deal with outliers [7, 84, 85]. All of these approaches have some primary assumptions such as; the percentage of available outliers in the dataset, sensitivity to outliers, and etc. To the best of authors knowledge, the least square approaches are not investigated in details for outlier detection in fuzzy linear regression. In this thesis, a two stage approach is proposed where in the first stage the outliers are detected and eliminated and then in the second stage a least squares approach is applied to fit a model to the clean dataset. In detecting the outliers, first we fit a model to the original dataset and then the observations which produce high error values are detected as ouliers and excluded from the dataset to make the clean dataset. Then in the second stage, a model is sought which minimizes the chosen model fitting measure over the cleaned dataset. The minimizition process is carried out with hybrid optimization based on tabu search and Nelder-Mead. The application of the hybrid optimization gives us the flexibility in the selection of any type of model fitting measurements based on the clients' requirements or the nature of dataset without worrying about the continuity or differentiability of the measure. The simulation results on the available datasets in the literature show the performance of the approach for detecting the outliers and ultimately to find the optimized model which can fit the genuine data.

6.1.1 Outlier detection approaches

In case of having outliers, the model obtained in (3.9) and (3.10) fits both legitimate and ilegitimate (outliers) data. So, the outliers should be detected and eliminated from the original dataset in the first stage and then in the second stage we seek for a model which only fits the clean dataset. The main approaches proposed to handle outliers in fuzzy linear regression are given by Peters [85], Lee and Yang [84], Chen [7] and Hung and Yang [83], which are probablistic aproaches.

Peters modified the approach by Tanaka et al. [5] to detect the outliers for crisp input and crisp output data by linear programming [85]. The linear program in Peters' approach maximizes the λ value which is in the interval $[0,1]$. The greater the λ value the better the quality of the data. However, in this approach, there are three user defined variables: d_0, p_i and p_0. The linear programming model is given as follows (6.1):

$$\begin{aligned}
&\max \lambda, \\
&\quad \text{subject to:} \\
&\quad (1-\lambda)p_0 - \sum_{i=1}^{N} c^t|x_i| \geq -d_0 \\
&\quad (1-\lambda)p_i + \alpha^t x_i + (1-h)c^t|x_i| \geq y_i \qquad (6.1)\\
&\quad \alpha^t x_i + (1-h)c^t|x_i| - (1-\lambda)p_i \leq y_i \\
&\quad 0 \leq \lambda \leq 1, c \leq 0, i = 1, 2, \cdot, N
\end{aligned}$$

A small d_0 value means that we are interested in a function with smaller spreads. Peters actually suggested $d_0 = 0$ which means the best function is a crisp function which ideally is rational as both inputs and outputs are crisp. Also p_i and p_0 are user defined values which are context-dependent. For example, if a dataset is assumes to have a lot of outliers then one may use greater p_i and p_0 values.

Chen proposed a probabilistic approach based on the linear program given by Tanaka et al. [5] to detect the outliers for crisp input-fuzzy output datasets [7]. The main idea of this approach is to keep the difference between the spreads of the estimated outputs and the actual ones less than a user defined value like $k \epsilon \mathbb{R}$. The linear program of this approach is given in (6.2). In this approach, by setting k as a large value, the model becomes less sensitive/affected to the outliers. In this model, e_i represents the spreads of the observed outputs.

$$\begin{aligned}
&\min J = \sum_{i=1}^{N} c_0 + c_1|x_{1i}| + \cdots + c_n|x_{ni}|, \\
&\quad \text{subject to:} \\
&\quad a^t x_i + (1-H)c^t|x_i| \geq y_i + (1-H)e_i \qquad (6.2)\\
&\quad a^t x_i - (1-H)c^t|x_i| \geq y_i - (1-H)e_i \\
&\quad c^t|x_i| - e_i \leq k, i = 1, 2, \cdot, N
\end{aligned}$$

Unlike the approaches proposed by Chen [7] and Peters [85], Hung and Yang [83] proposed an outlier omission approach which does not have any user defined variables as the ones in Chen's or Peters' approaches. However, Hung and Yang's approach assumes that there exists at least one outlier in the data set. So, the procedure is to first delete the ith observation and then apply the probabilistic approach as given by Tanaka et al. [5]. The same approach is applied for the $(n-1)$ remaining observations and the models' errors are stored O_i. If O_0 is the minimized value of the objective function in the presence of all observations, then the model with the smallest $d_i = |O_i - O_0|, i = 1, \cdots, n$ is detected as the best model and the observation i^{th} is considerd to be the outlier. In case there are two or more outliers in the dataset, the same approach should be taken with eliminating all the combinations of two or more observations. However, the process becomes computationally expensive as for example in case of having 100 observations and 10 outliers we need to have $\binom{100}{10} = \frac{10!}{90!10!} \simeq 17e12$. Morever, we need to know how many outliers actually exist in the dataset, otherwise, the problem becomes even more computationally expensive.

6.1.2 Outlier detection with global continuous optimization

All the proposed approaches for the detection of outliers in fuzzy linear regression in the literature; Peters [85], Chen [7], and Hung and Yang [83], are improvements on the probabilistic approaches. The goal of probablistic approaches is to cover the spreads of the outputs as much as possible. However, least square based approaches are the generalized form of statistical linear regression analysis and their goal is to find a model by minimizing a model fitting measurement value. The motivation of the proposed method are as follows;

1. to eliminate the user defined variables, a problem which makes the aproaches in [84, 85, 107] context-dependent, and
2. to be able to eliminate more than one outlier without having a computationally expensive process– a problem which exists in Hung and Yang's approach [83].

The proposed method is carried out in two stages. In the first stage, a model with a minimum of model fitting measure is sought over the original dataset. Then in the second stage, the observations which have high model fitting measures are detected as outliers and so excluded from the dataset, as a result we have a clean dataset. The obtained model in the second stage does not fit the outliers ppresented in the original dataset. The algorithm of the proposed method is given in Algorithm 1.

Algorithm 6.1 Proposed algorithm for fuzzy linear regression with outlier elimination

```
stage 1:
Seek the best model, $\widetilde{f}$ with total minimum error of the chosen model fitting measurement $O_{\widetilde{f}}$ for the *original dataset*,
Calculate the error of the chosen model fitting measurement for each observation $O^i{}_{\widetilde{f}}$, and the average error of $\overline{O^i{}_{\widetilde{f}}} = \sum_{i=1}^{n} \frac{O^i{}_{\widetilde{f}}}{n}$
**for** $i = 1$ to $n$ **do**
    **if** $O^i \geq \overline{O^i{}_{\widetilde{f}}}$ **then**
        *clean dataset* $\Leftarrow$ exclude the $i^{th}$ observation from the dataset
    **end if**
**end for**
stage 2:
Seek the best model, $\widetilde{f}^*$ with total minimum error of the chosen model fitting measurement $O_{\widetilde{f}^*}$ for the clean dataset
```

If all the outliers are correctly eliminated in the first stage, then the standard deviation *std* of all observations' errors for the *clean dataset* is expected to be very close to zero in the second stage. However, a large *std* value in the second stage can be due to either incorrect outlier detection, or non-linearity of the dataset.

There are different model fitting measurements in the literature of fuzzy linear regression such as Euclidean distance [145], distance criterion [146], compatibility measure [102], relative none-intersected area [12], and Hojati's measure [4] and etc. In this thesis, we use Hojati's measure given in (6.3) due to the fact that this measurement is in the range of 0 to 1 which makes it easier for interpretation. Majority of the proposed measurements range from 0 to $+\infty$. In (6.3), $\mu_{\widetilde{y}^*}$ and $\mu_{\widetilde{y}}$ are the membership functions of the observed and the estimated outputs, respectively. And $S_{\widetilde{y}_i^*}$ and $S_{\widetilde{y}_i}$ are the supports of the observed output, $\widetilde{y}_i^*$, and the estimated output, $\widetilde{y}_i$, respectively. Note that in our proposed method, any other model fitting measurement can also be applied based on the clients' requests and requirements or the nature of the dataset.

$$S_h = \frac{\int_{S_{\widetilde{y}_i^*} \bigcup S_{\widetilde{y}_i}} \min(\mu_{\widetilde{y}^*(x)}, \mu_{\widetilde{y}(x)})dx}{\int_{S_{\widetilde{y}_i^*} \bigcup S_{\widetilde{y}_i}} \max(\mu_{\widetilde{y}^*(x)}, \mu_{\widetilde{y}(x)})dx} \tag{6.3}$$

To find the best model for a data set which minimizes the model fitting measurement given in (6.3), a hybrid optimization method based on tabu search and Nelder-Mead approach is applied [6, 89]. Tabu search is a powerful search method for diversification as it escapes from the local minima by keeping a record of the searched areas thus avoiding re-searching them. Tabu search is a point to point search strategy which also makes it less computationally expensive in terms of the number of function evaluations for few-dimensional optimization problems compared to population-based approaches such as genetic algorithms or particle swarm optimization [6, 16]. However, global optimization approaches are very slow if a very close approximation of the global minimum solution is sought. That's the motivation of applying a local optimizer method such as Nelder-Mead to receive the solution from the diversification phase for further optimization. The hybrid optimization method based on tabu search and Nelder-Mead proposed by Mashinchi et al. is reported to be superior or Pareto equivalent to other methods for small dimensional optimization problems [2]. We have applied the same specification of the hybrid optimization approach given in [6] to seek the optimized model in both stages 1 and 2 of Algorithm 6.1.

As mentioned earlier, a triangular fuzzy number can be represented by three parameters. So, for an n-dimensional data set, we search for model given in (3.10) which has $3n+3$ parameters to be optimized. The formulation of the optimization problem is as follows [89]:

$$\min\left(\sum_{i=1}^{n} \frac{\int_{S_{\widetilde{y}_i^*} \bigcup S_{\widetilde{y}_i}} \min(\mu_{\widetilde{y}_i^*(x)}, \mu_{\widetilde{y}_i(x)})dx}{\int_{S_{\widetilde{y}_i^*} \bigcup S_{\widetilde{y}_i}} \max(\mu_{\widetilde{y}_i^*(x)}, \mu_{\widetilde{y}_i(x)})dx}\right) \tag{6.4}$$
$$f : \mathbb{R}^{3n+3} \to \mathbb{R}$$

6.1.3 Results

To validate the proposed approach we have used three datasets. The first two datasets used by [7,83,84] are small datasets with constant, and increasing spreads. The outliers are just assumed to occur in the spreads and not in the centers of the data. The other dataset which is produced by an approach given by [147] is large and has 5 percent random outliers both in centers and the spreads of the data. We have run the optimization approach for 100 separate runs and reported the average results. All the simulations were run on PentiumIV 3.00GHz with 2GB memory. The program is coded in Matlab 7.0. The standard deviation, average number of function evaluations, and the average execution time are denoted by std, F_e, and E_t,

Table 6.1: Constant, increasing and decreasing spreads datasets with existence of outliers (taken from [8])

x	Dataset 1			Dataset 2		
	Output	1^{st} st. error	2^{nd} st. error	Output	1^{st} st. error	2^{nd} st. error
1	(8.0,1.8)	0.99^d	-	(11,2)	0.00	0.00
2	(6.4,2.2)	0.02	0.0572	(13,2)	0.88^d	-
3	(9.5,2.6)	0.50^d	-	(21,4)	0.23	0.23
4	(13.5,2.6)	0.89^d	-	(29,4)	0.78^d	-
5	(13,2.4)	0.02	0.0590	(29,6)	0.24	0.24
6	(15.2,2.3)	0.05	0.0223	(34,6)	0.19	0.18
7	(17.0,2.2)	0.24	0.2845	$(45,15)^o$	0.60^d	-
8	$(19.3,4.8)^o$	0.48^d	-	(44,8)	0.17	0.16
9	(20.1,1.9)	0.78^d	-	(48,12)	0.34	0.13
10	(24.3,2.0)	0.29	0.2251	(54,12)	0.10	0.10
std	-	0.28	0.1176	-	0.30	0.08
F_e	-	437	433	-	463	379
E_t	-	2.19s	1.16s	-	2.51s	1.85s
Ave_e	-	0.49	0.31	-	0.37	0.22

^{o} Represents the outlier data.
^{d} Represents the detected outliers in the first stage.

respectively.

6.1.3.1 small dataset with outliers happening in the spreads

These three datasets are originally proposed by Tanaka et al. [5] and modified by [84] to introduce outliers. This dataset is then used by [7, 83, 84] for the validation of their proposed approaches. There is only one outlier in each dataset and the outlier happens just in the spreads.

The best models for all three data sets, which are used to eliminate the outliers, detected the outliers correctly. The experimental results and the datasets are given in Table 1. Out of 100 separate runs, in 54, and 97 runs the set of detected outliers include the the actual outlier for dataset 1, 2 respectively. As expected, the *std* of the error for the second stage for all the datasets is very close to zero. This means that the outliers are correctly eliminated and the datasets have linear trends.

The comparison of the results obtained from our approach to the one given by Chen [7] and Lee and Chang [84] is given in Table 6.3. The estimated models from this work and the one proposed by Chen [7] is given in Figure 6.1. According to

Figure 6.1 Chen's approach has wider spreads to cover all the data. As a common property of the probabilistic approaches, with more observations the spreads of the model get wider. This makes probabilistic approaches to be more sensitive to outliers [148].

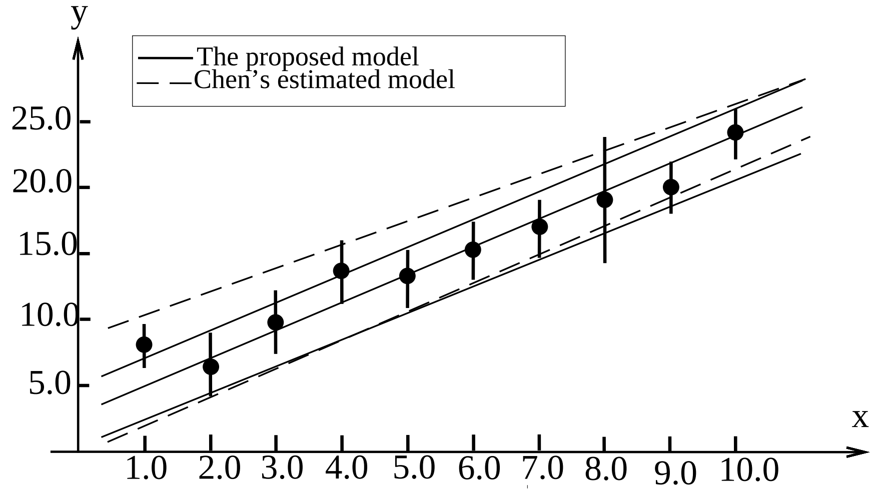


Figure 6.1: Comparison between Chen's estimated model [7] and our approach over Dataset 1 (taken from [8])

6.1.3.2 large dataset with outliers happening in both centers and spreads

To test the perfomance of the proposed method for a large data set and outliers, we applied the same approach and function as given by Buckley and Feuring [147]. The linear function is given as $\widetilde{y} = (5.0, 5.2, 5.7)\widetilde{x} + (3.0, 3.2, 3.7) + \epsilon$, where in 95 percent of the time $\epsilon = RND[-0.5, 0.5]$ and 5 percent of the time $\epsilon = RND[-5, -2]$ or $RND[2, 5]$ is used to generate outputs from 1,000 randomly chosen $\widetilde{x}$ in [1,1000]. Then we attempt to fit $\widetilde{y} = \widetilde{k_1}\widetilde{x} + \widetilde{k_2}$ to the generated asymmetric fuzzy input-asymmetric fuzzy output with outliers. Note that the data set is generated once, however, we have run the method for 1,000 times.

Selected fitness measure is the hybrid distance measure given in (4.5) [6] The average errors of RNIA, NIA Hojati's measurement, and distance criterion for the best obtained model with $\widetilde{k_1} = (5.009, 5.488, 5.699)$ and $\widetilde{k_2} = (2.866, 3.062, 4.053)$ are 38.1736, 5.9010e+003, 0.0355, and 4.3985e+003, respectively. The average RNIA, CPU time and standard deviations of RNIA and CPU time are 253.73, 66.4810 seconds, and 116.7378 and 6.48 seconds, respectively. For the best obtained model, the NIA between the original output and the ideal output and the estimated output and the ideal output for the outliers are represented by outliers' error and estimated error in Table 6.2. The ideal outputs are obtained from $\widetilde{y} = (5.5, 5.5, 5.7)\widetilde{x} + (3.0, 3.2, 3.7)$ without any noise. In Table 6.2, 25 out of 42 (59%) of the estimated outputs generate less error than the outliers' error. In other words, 59% of the outliers are detected and the obtained model is not fitting them. However, one may design and apply a specific objective function to be minimized for data sets with outliers. As mentioned earlier, in this thesis we used the objective function given in (4.5) which is not necessarily the best objective function for the detection of outliers.

Table 6.2: The estimated error (4.9) of the best obtained model for the outliers in the large dataset (taken from [8])

No.	Ins.	Input	Ideal output	Outlier's output	Est. by the model	Out.'s' err.	Est. err.
1	50	(577.9,580.6,583.2)	(2892.7,3022.4,3328.0)	(2900.9,3205.1,3336.2)	(2898.0,3189.7,3328.0)	136.2	123.1
2	55	(127.6,129.5,131.1)	(640.7,676.7,751.0)	(646.8,722.0,757.1)	(641.8,713.9,751.2)	37.7	28.4
3	81	(789.0,789.3,791.2)	(3948.1,4107.4,4513.4)	(3938.6,4335.0,4503.9)	(3955.4,4334.8,4513.2)	154.3	165.1
4	103	(222.0,222.1,224.2)	(1113.2,1158.2,1281.7)	(1104.7,1216.6,1273.2)	(1115.2,1222.1,1281.9)	37.6	47.3
5	108	(425.2,425.3,427.6)	(2128.9,2214.8,2440.9)	(2120.8,2334.5,2432.7)	(2132.8,2337.3,2441.0)	80.0	89.6
6	116	(118.6,121.2,121.4)	(595.8,633.5,695.9)	(588.1,662.4,688.2)	(596.8,668.3,696.1)	18.4	26.4
7	181	(9.8,11.4,11.8)	(51.8,62.3,71.2)	(58.8,73.0,78.2)	(51.8,65.5,71.5)	12.7	2.9
8	184	(873.4,875.6,876.7)	(4369.8,4556.3,5001.1)	(4361.1,4810.5,4992.3)	(4378.0,4808.5,5000.8)	173.8	183.5
9	204	(273.1,273.9,276.5)	(1368.3,1427.5,1579.7)	(1362.1,1503.8,1573.5)	(1370.8,1506.3,1579.8)	51.2	58.5
10	220	(820.8,823.6,826.1)	(4107.0,4286.0,4712.7)	(4097.5,4523.9,4703.2)	(4114.7,4523.3,4712.5)	162.8	173.6
11	226	(482.9,485.5,485.6)	(2417.6,2527.8,2771.7)	(2410.5,2666.5,2764.5)	(2422.1,2667.6,2771.7)	93.8	102.1
12	242	(463.5,466.3,469.0)	(2320.5,2428.0,2676.9)	(2330.2,2577.9,2686.6)	(2324.7,2562.3,2676.9)	114.4	99.4
13	246	(494.1,495.3,497.8)	(2473.6,2578.8,2840.9)	(2465.5,2719.5,2832.8)	(2478.2,2721.5,2840.9)	95.1	104.7
14	272	(88.2,91.0,92.7)	(444.1,476.2,532.2)	(437.9,497.6,526.0)	(444.8,502.3,532.5)	14.0	20.6
15	278	(734.2,736.6,739.5)	(3673.8,3833.3,4218.8)	(3679.5,4060.2,4224.4)	(3680.7,4045.5,218.6)	165.4	155.5
16	280	(288.6,290.9,293.4)	(1446.0,1515.8,1676.2)	(1439.8,1597.2,1670.0)	(1448.6,1599.6,1676.4)	55.2	62.6
17	295	(408.4,410.0,410.7)	(2045.2,2135.2,2344.5)	(2054.7,2268.1,2354.1)	(2048.9,2253.2,2344.6)	100.7	86.3
18	315	(957.6,958.0,960.3)	(4791.1,4985.0,5477.2)	(4785.8,5267.5,5471.9)	(4800.0,5261.1,5476.9)	195.5	200.4
19	323	(38.6,41.2,44.0)	(195.9,217.6,254.5)	(190.4,224.7,249.0)	(196.1,229.3,254.8)	5.7	10.1
20	350	(56.8,58.5,60.6)	(286.8,307.5,348.8)	(293.2,331.8,355.3)	(287.2,324.3,349.1)	23.48	13.5
21	407	(882.4,884.3,885.2)	(4415.1,4601.5,5049.4)	(4420.2,4872.1,5054.4)	(4423.3,4856.2,5049.1)	194.3	1185.1
22	507	(930.0,932.2,933.7)	(4652.9,4850.6,5326.0)	(4646.1,5123.7,5319.1)	(4661.6,5119.2,5325.7)	188.3	195.5
23	561	(6.5,7.0,7.8)	(35.5,39.8,47.9)	(44.8,51.5,57.1)	(35.5,41.7,48.2)	11.7	1.8
24	604	(333.0,333.1,336.7)	(1667.9,1738.8,1922.6)	(1660.2,1831.6,1915.0)	(1670.9,1834.9,1922.7)	62.1	71.1
25	629	(391.4,393.8,396.3)	(1960.0,2051.2,2262.8)	(1968.4,2178.0,2271.2)	(1963.6,2164.6,2262.9)	97.1	84.1
26	653	(662.3,664.2,667.2)	(3314.6,3457.0,3806.5)	(3323.7,3665.7,3815.6)	(3320.8,3648.4,3806.4)	154.8	140.3
27	675	(15.3,17.7,18.5)	(79.3,95.3,109.0)	(74.8,96.5,104.5)	(79.3,100.3,109.4)	3.8	4.4
28	716	(66.1,67.8,68.6)	(333.0,355.7,394.7)	(325.1,368.4,386.8)	(333.5,375.1,395.0)	8.9	15.1
29	763	(730.4,733.2,735.8)	(3654.9,3816.0,4197.7)	(3645.0,4026.4,4187.8)	(3661.7,4027.3,4197.6)	143.4	154.8
30	770	(294.1,296.5,297.0)	(1473.6,1544.9,1696.9)	(1467.3,1627.7,1690.5)	(1476.3,1630.2,1697.0)	55.4	62.9
31	775	(331.0,331.9,333.3)	(1657.7,1728.8,1903.6)	(1653.0,1824.0,1898.9)	(1660.7,1824.4,1903.7)	64.7	70.1
32	787	(696.0,696.0,698.9)	(3483.2,3622.7,3987.7)	(3489.5,3838.1,3994.0)	(3489.7,3823.2,3987.6)	156.7	146.1
33	797	(542.3,544.3,545.2)	(2714.5,2833.4,3111.5)	(2705.9,2988.5,3102.9)	(2719.5,2990.2,3111.4)	104.5	114.5
34	820	(964.3,966.4,968.6)	(4824.3,5028.4,5524.7)	(4833.3,5327.6,5533.7)	(4833.3,5306.8,5524.3)	217.9	202.8
35	836	(199.8,202.1,203.0)	(1001.9,1054.1,1161.1)	(1007.5,1120.6,1166.7)	(1003.7,1112.2,1161.3)	52.0	43.4
36	871	(31.4,33.3,35.5)	(159.8,176.6,205.9)	(150.3,177.4,196.5)	(160.0,186.0,206.3)	8.8	8.1
37	875	(224.8,226.5,227.9)	(1127.0,1181.1,1302.9)	(1120.5,1242.9,1296.4)	(1129.0,1246.3,1303.1)	40.9	48.5
38	898	(654.0,654.6,655.5)	(3273.1,3406.9,3740.0)	(3279.6,3609.9,3746.4)	(3279.2,3595.4,3739.9)	147.3	136.8
39	899	(616.6,619.5,620.4)	(3085.8,3224.6,3539.8)	(3079.9,3404.9,3533.9)	(3091.5,3403.1,3539.7)	124.0	130.5
40	955	(516.7,519.4,521.4)	(2586.7,2704.2,2975.5)	(2578.7,2852.3,2967.5)	(2591.4,2853.8,2975.4)	100.7	110.0
41	963	(730.1,730.5,732.4)	(3653.6,3801.9,4178.6)	(3661.6,4029.3,4186.6)	(3660.4,4012.4,4178.5)	165.9	153.0
42	985	(531.5,533.8,535.4)	(2660.5,2779.1,3055.5)	(2651.1,2930.2,3046.1)	(2665.4,2932.9,3055.5)	101.7	112.7

Selected fitness measure is Hojati's distance measure given in (6.4) [4] We attempt to fit $\widetilde{y} = \widetilde{k_1}\widetilde{x} + \widetilde{k_2}$ to the generated asymmetric fuzzy input-asymmetric fuzzy output with outliers. The average error of the obtained models for 100 separate runs in the first stage is 0.0976 with the best model of $\widetilde{y} = (5.01, 5.47, 5.69)\widetilde{x} + (-0.09, 5.27, 6.81)$. In the first stage, 30 outliers out of 42 are correctly detected and eliminated from the dataset. The average number of iterations and execution time are 365 and 218s, respectively. With half iterations are devoted to diversification by tabu search with 10 neighbors and half to indetsification. In the second stage, the best model which fits the *clean dataset* is $\widetilde{y} = (4.99, 5.49, 5.68)\widetilde{x} + (2.8, 3.21, 4.48)$ with model fitting measure of 0.0650 over the actual dataset. The average number of iterations and execution time in the second stage are 378.3 and 214.06, respectively. With half iterations are devoted to diversification by tabu search with 10 neighbors and half to indetsification. Note that even if we know for example 50 data items are outliers in a dataset with the size of 1000 data items, then the approach given by Hung and Yang [83] needs to solve $\binom{1000}{50} = \frac{1000!}{950!50!}$ linear programs.

Table 6.3: Comparison between different methods for the Datasets 1 and 2 (taken from [8])

Dataset	Approach	Model	Ave. MFM given in (6.3) (actual outlier excluded)
Dataset 1	This work	$\widetilde{y} = (-0.3824, 1.9479, 4.1347)$ $+(2.2111, 2.2170, 2.2266)x$	0.42
	Chen's [7]	$\widetilde{y} = (4.75, 4.55) + (1.85. - 0.15^{*})x$	0.62
	Lee's [84]	$\widetilde{y} = (4.43, 3.76) + (1.86, 0.14)x$	0.57
Dataset 2	This work	$\widetilde{y} = (5.24, 6.37, 7.12) + (3.75, 4.62, 5.87)x$	0.30
	Chen's [7]	$\widetilde{y} = (5.76, 0.90) + (4.95, 1.38)x$	0.40
	Lee's [84]	$\widetilde{y} = (4.51, 0.65) + (5.70, 2.13)x$	0.59

[*] γ cannot be negative in $(c, \gamma)_{LR}$ representation. So we assume it should be positive.

6.2 Spread increasing problem

Some fuzzy linear regression models constantly increase the spreads of the outputs with the increase in the magnitude of the inputs. This problem is known as the Spread Increasing Problem (SIP) [86]. For introducing the fuzzy linear regression analysis approach based on global continuous optimization, some definitions are provided first.

Definition 10 (Magnitude comparison)**.** *A vector like $\overrightarrow{x_j} = \langle x_j^1, x_j^2, \cdots, x_j^n \rangle$ where $x_j^k \epsilon \mathbb{R}$ is said to have a larger magnitute compared to a vector like $\overrightarrow{x_i} = \langle x_i^1, x_i^2, \cdots, x_i^n \rangle$ where $x_i^k \epsilon \mathbb{R}$, if there is at least one $g \epsilon \{1, 2, 3, \cdots, n\}$ where $x_j^g > x_i^g$ and for all $e \neq g$, $x_j^e \leq x_i^e$. It is shown by $\overrightarrow{x_i} << \overrightarrow{x_j}$.*

Definition 11 (Pareto equivalent magnitudes)**.** *Vectors $\overrightarrow{x_j} = \langle x_j^1, x_j^2, \cdots, x_j^n \rangle$ and $\overrightarrow{x_i} = \langle x_i^1, x_i^2, \cdots, x_i^n \rangle$ are Pareto equivalent in terms of their magnitude if neither $\overrightarrow{x_i} << \overrightarrow{x_j}$ nor $\overrightarrow{x_i} >> \overrightarrow{x_j}$ holds true.*

Note that in both the above definitions, only crisp values are considered for magnitude comparison. However if the dataset has fuzzy number input variables, like $\langle \widetilde{x_i^1}, \widetilde{x_i^2}, \cdots, \widetilde{x_i^n} \rangle$, then one can defuzzify the input variables to be able to compare the magnitudes.

As an example let us assume we have three vectors $\overrightarrow{x_1} = \langle 1, 2, 1, 3, 5 \rangle$, $\overrightarrow{x_2} = \langle 1, 1, 1, -3, 6 \rangle$, and $\overrightarrow{x_3} = \langle 1, 2, 2, 3, 7 \rangle$. According to the above definition $\overrightarrow{x_1}$ and $\overrightarrow{x_2}$ are *Pareto equivalent* as $x_1^2 > x_2^2$ and $x_1^5 < x_2^5$, but $\overrightarrow{x_1} << \overrightarrow{x_3}$ and $\overrightarrow{x_2} << \overrightarrow{x_3}$ as $\{x_1^1 = x_3^1,\ x_1^2 = x_3^2,\ x_1^3 < x_3^3,\ x_1^4 = x_3^4,\ x_1^5 < x_3^5\}$ and $\{x_2^1 = x_3^1,\ x_2^2 < x_3^2,\ x_1^3 < x_3^3,\ x_1^4 = x_3^4,\ x_1^5 < x_3^5\}$, respectively.

Definition 12 (Definition: SIP-free fuzzy regression)**.** *For a given data set, a fuzzy triangular linear function is SIP-free if there exits at least one i and j where $\overrightarrow{x_i} << \overrightarrow{x_j}$, and $s(\widetilde{y_i}) > s(\widetilde{y_j})$ where $s(\widetilde{y})$ returns a real value representing the spread of $\widetilde{y}$.*

However, if the above condition does not hold, one *may not* conclude that the model has SIP, as sometimes the nature of the dataset dictates the model to have constantly increasing spreads to fit the output data.

6.2.1 Anti-spread increasing approaches

Kao and Chyo [88] proposed an approach which seeks a model in the form of $\widetilde{y} = Ax + B + \widetilde{C}$ (or $\widetilde{y} = A\widetilde{x} + B + \widetilde{C}$ depending on the type of input data) with crisp coefficients and an extra fuzzy constant. The extra fuzzy coefficient $\widetilde{C}$ is introduced in the model so that it can generate fuzzy output from crisp input. This model does not have SIP as regardless of the inputs magnitude the outputs have the same spreads which are equal to the spread of $\widetilde{C}$. However, the defficiency of this approach is that the sought models do not properly fit the data with a decreasing/increasing nature. The model is represented by few number of parameters which makes the model easy to interpret. In case of having asymmetric triangular numbers, the model has only $m+3$ parameters where m is the dimension of the input variable.

Nasrabadi and Nasrabadi [136] tackled SIP by introducing a new fuzzy multiplication operation. Based on their defined operator, the result of multiplying two fuzzy numbers such as $\widetilde{A} = (m_A, \alpha_A)$ and $\widetilde{B} = (m_B, \alpha_B)$ is $\widetilde{A} \otimes \widetilde{B} = (m_A \times m_B, \alpha_A \times \alpha_B)$. Applying this algebraic operation solves the SIP, as the spreads of the output are not dependent on the magnitude of the input and only dependent on the spreads of the inputs. However, the defficieny of this model is that the spreads of the output are just dependent on the spreads of the inputs. So, if there is any relationship between the spreads of the output and the magnitude of the inputs, it cannot be captured by this model [86]. The model in this approach is in form of

$$\begin{aligned} \widetilde{y_i} &= \widetilde{A}_n \otimes \widetilde{x}_{in} + \widetilde{A}_{n-1} \otimes \widetilde{x}_{in-1} + \cdots + \widetilde{A}_1 \otimes \widetilde{x}_{i1} + \widetilde{A}_0 \\ &= \left(m_{A_0} + \sum_{j=1}^{n} m_{A_j} m_{x_ij}, \alpha_{a_0} + \sum_{j=1}^{n} \alpha_{A_j} \alpha_{x_ij} \right) \end{aligned} \tag{6.5}$$

The above model has $3n+3$ parameters, where n is the dimension of the input variable. The disadvantage of this model is that if the input variables are crisp, then the model's spread is constant and equal to α_{A_j}.

In another approach, Modaress et al. [135] fit the data with a model in the form of $\widetilde{y} = (m_{A_n} m_{x_n} + m_{A_{n-1}} m_{x_{n-1}} + \cdots + m_{A_1} m_{x_1} + m_{A_0}, \alpha A_n \alpha m_n + \alpha A_{n-1} \alpha m_{n-1} + \cdots + \alpha A_1 \alpha m_1 + \alpha A_0)$. In this model α_{A_i} and α_{m_i} which are the spreads of the coefficients and the inputs, are allowed to be negative, positive or zero. With the ability to have some negative α_{m_i}, the model can have decreasing spreads. The LR-type of this model has $3n+3$ variables. The deficiency of this model is that it can only produce decreasing, increasing or constant spreads at a time but variable spreads.

Chen and Dang [87] introduced a fuzzy error notation in their model. Their regression model is in the form of $\widetilde{y} = A_n x_n + A_{n-1} x_{n-1} + \cdots + A_n x_n + A_1 x_1 + A_0 + \widetilde{e}$, where $\widetilde{e} = (e_l, 0, e_r)$. The objective of this approach is to minimize the difference between the actual and the estimated fuzzy output. The model from this appoach has $n+2m+1$

parameters which means the number of variables is proportional to the number of instances. The dependency between the number of parameters representing the model and the number of input variables makes the model very complicated to be interpreted, if the number of input variables increases. Another issue with the model obtained by this approach is that the model cannot produce fuzzy output for a crisp input data. As an example, if the regression model is $\widetilde{y_i} = 2x_i + 3 + \widetilde{e_i}$, then for a new datum like $x = 5$ as the $\widetilde{e_i}$ is unknown, the output will be $y_i = 2 \times 3 + 3$.

In a recent approach, Lu and Wang [86] proposed a fuzzy linear regression model in which in contrast to the Chen and Dang's model [87], there is no fuzzy error term for each observation in the model. Not having the fuzzy error term, makes the number of parameters of the model to be independent of the number of observations. Moreover in contrast to the Kao Chyu's models in [88], the spreads of the estimated output in Lu and Wang's model is also dependent on the input variable. So, if there is any relationship between the spreads of the output and the input values, it can be captured. The proposed model for multiple independent variables $\widetilde{X_{ij}} = (m_{x_{ij}}, \alpha_{x_{ij}}, \beta_{x_{ij}})_{LR}$ is given as:

$$\begin{aligned}
&\widetilde{y_i} = A_n m_{x_{in}} + An - 1 m_{x_{in-1}} + \cdots + A_1 m_{x_{i1}} + A_0 + \widetilde{S_i}\\
&\widetilde{S_i} = (0, \alpha_{S_i}, \beta_{S_i})\\
&\alpha_{S_i} = \sum_{t=1}^{m} A_{llt}\alpha_{x_{it}} + \sum_{t=1}^{m} A_{lmt}\alpha_{x_{it}} + \sum_{t=1}^{m} A_{lrt}\alpha_{x_{it}} + c_l\\
&\beta_{S_i} = \sum_{t=1}^{m} A_{rlt}\alpha_{x_{it}} + \sum_{t=1}^{m} A_{rmt}\alpha_{x_{it}} + \sum_{t=1}^{m} A_{rrt}\alpha_{x_{it}} + c_r\\
&\alpha_{S_i} \geq 0, \beta_{S_i} \geq 0\\
&\widetilde{x_i} = m_{x_i}, \alpha_{x_i}, \beta_{x_i}
\end{aligned} \tag{6.6}$$

where $\widetilde{S_i}$ is the spreads of the output for the instance i, and consists of A_{llt}, A_{lmt}, A_{lrt}, A_{rlt}, A_{rmt}, A_{rrt}, and c_r. This model results in an optimization problem with $7m + 3$ variables which is solved with an iterative approach [86]. Since this approach applies an iterative optimization method, if the dimension of input variables increases then it becomes computationally expensive to derive the regression model [86].

The summay of the studies in [86–88, 135, 136] is given in Table 6.4.

Table 6.4: The summary of the studies to tackle SIP (taken from [9])

Approach given by	Properties				
	Constant spreads	Decreasing spreads	Variable spreads	Num. of param. LR-type	Num. of param. in L-type
Lu and Wang [86] (2009) [1]	Yes	Yes	No	$7n+3$	$4n+2$
Chen and Dang [87] (2008) [2]	Yes	Yes	Yes	$n+2m+1$	$n+m+1$
Modaress et al. [135] (2005) [3]	Yes	Yes	No	$3n+3$	$2n+2$
Nasrabadi and Nasrabadi [136] (2009) [4]	Yes	Yes	No	$3n+3$	$2n+2$
Kao and Chyu [88] (2002)[5]	Yes	No (crisp input) Yes (fuzzy input)	No	$n+4$	$n+2$

[1] The relationship between the spreads of outputs and the magnitude of the inputs is captureed-if there is any

[2] Each input has its own fuzzy error

[2] The model cannot produce fuzzy output for a new crisp data

[2] The model is hard to interpret due to large number of parameters

[3] For a fuzzy value like $\widetilde{x} = (m_x, \alpha_x)$, it is allowed that $\alpha_x \epsilon \mathbb{R}$

[4] A new algebraic multiplication operation is introduced

[4] The relationship between the magnitude of the inputs and the spreads of the outputs cannot be captured

[5] The model is easy to be interpreted due to few number of parameters

6.2.2 Drawbacks of the existing methods

In tackling SIP, all the proposed approaches in [86, 88, 135, 136], except the one proposed by Chen and Dang [87], can only estimate models with either decreasing, increasing or constant spreads at a time. The model proposed by Chen and Dang has this interesting property which can produce variable spreads. In other words, for some intervals of the input variables, the regression model has decreasing spreads while for some other intervals the model may have constant or increasing spreads depending on the nature of the input-output data set. However, if the Chen and Dang's model receives a new crisp input, it cannot produce fuzzy output. This is due to the fact that in their apporach, the only fuzzy term is the error term which is different for each of the inputs, and this error term is undefined for a new crisp input. Moreover, the number of parameters of this model to be optimized is proportional to the number of instances [86, 123], see Table (6.4). On the other hand, although the model proposed by Lu and Wang [86] does not have SIP, it cannot support variable spreads. The number of parameters to be optimized by this approach is independent of the number of inputs. However, this approach is still computationally expensive comparing to Chen and Dang's approach for few inputs but with high dimension, see Table (6.4).

In industrial applications, models with many parameters are hard to justify as the interpretation of the model becomes more complex. Moreover, if an iterative process is applied for deriving such models, the increase in the number of parameters makes the process more time consuming. The number of parameters in other SIP-free approaches proposed by Modarss et al. [135], Nasrabadi and Nasrabadi [136] and Kao and Chyu [88] are less than the ones proposed by Chen and Dang [87] and Lu and Wang [86]. The models proposed by Modarss et al. [135], and Nasrabadi and Nasrabadi [136] can still be improved to fit the data with a smaller error, and the model proposed by Kao and Chyu [88] only supports constant spreads.

This thesis proposes an SIP free fuzzy linear regression model which has the following properties:

- a fuzzy linear model with a fewer number of parameters in comparison to Chen and Dang's [87], and Lu and Wang's [86] and an equal number of parameters to the ones proposed by Modaress et al. [135] and Nasrabadi and Nasrabadi [136],

- it supports variable spreads, and

- the sought model based on the proposed approach is either superior or Pareto-equivalent to the models in the literature based on the chosen model fitting measurements.

6.2.3 Anti-spread increasing approach based on global continuous optimization

We assume a semi-linear model in the form of $\hat{y} = \hat{A}_n\hat{x}_n + \hat{A}_{n-1}\hat{x}_{n-1} + \cdots + \hat{A}_1\hat{x}_1 + \hat{A}_0$, where ˆ represents a variable with 3-tuple elements. For example the coefficient, $\hat{A}$ can be in the form of any 3-tuple like $\langle a^1, a^2, a^3 \rangle$ where a^1, a^2 and a^3 are in $\mathbb{R}$. The model can be expanded as:

$$\begin{aligned}(y^1,y^2,y^3) = (a_n^1, a_n^2, a_n^2) \times (x_n^1, x_n^2, x_n^3) + (a_{n-1}^1, a_{n-1}^2, a_{n-1}^3) \\ \times (x_{n-1}^1, x_{n-1}^2, x_{n-1}^3) + \cdots + (a_1^1, a_1^2, a_1^3) \times (x_1^1, x_1^2, x_1^3) + (a_0^1, a_0^2, a_0^3)\end{aligned} \tag{6.7}$$

By applying the following introduced algebraic calculations of two 3-tuples like (a^1,a^2,a^3) and (b^1,b^2,b^3) where all a^i and b^i are in $\mathbb{R}$,for $i = 1,2,3$, the model in (6.7) is converted to (6.8), where the operations are as:
Addition:
$(a^1,a^2,a^3) + (b^1,b^2,b^3) = (a^1 + b^1, a^2 + b^2, a^3 + b^3)$
Multiplication:
$(a^1,a^2,a^3) \times (b^1,b^2,b^3) = (a^1b^1, a^2b^2, a^3b^3)$

$$\begin{aligned}(y^1,y^2,y^3) =& (a_n^1x_n^1, a_n^2x_n^2, a_n^3x_n^3) + (a_{n-1}^1x_{n-1}^1, a_{n-1}^2x_{n-1}^2, a_{n-1}^3x_{n-1}^3) + \cdots + \\ & (a_1^1x_1^1, a_1^2x_1^2, a_1^3x_1^3) + (a_0^1, a_0^2, a_0^3) = \\ & (a_n^1x_n^1 + a_{n-1}^1x_{n-1}^1 + \cdots + a_1^1x_1^1 + a_0^1, \\ & a_n^2x_n^2 + a_{n-1}^2x_{n-1}^2 + \cdots + a_1^2x_1^2 + a_0^2, \\ & a_n^3x_n^3 + a_{n-1}^3x_{n-1}^3 + \cdots + a_1^3x_1^3 + a_0^3)\end{aligned} \tag{6.8}$$

The 3-tuple (y^1,y^2,y^3) given in (6.8) can represent a fuzzy triangular number if the three elements are in their ascending order. However, there is no gurantee that $y^1 < y^2 < y^3$. Thus by sorting y^1, y^2 and y^3, we would form a triangular fuzzy number. Basically all y^1, y^2 and y^3 are linear lines. However their roles in terms of representing the right, left or center of the fuzzy triangular number change with different inputs. So, for each input value, the y^i which has the larger value, represents the right spread and the smallest one represents the left spread and the remaining one is the center. For doing so a sorting operation *sort* is applied on 3-tuple to ascending order. Figure 6.2 illusterates an example of a fuzzy semi-linear model where the role of each y^1, y^2 and y^3 changes based on different inputs. The dashed lines in Figure 6.2 show the right, left and the center of the fuzzy number after sorting operation. For example y^3 is the right spread of the output for $x < a$, the center of the output for $a \leq x < b$, and the right spread of the output for $b \leq x$.

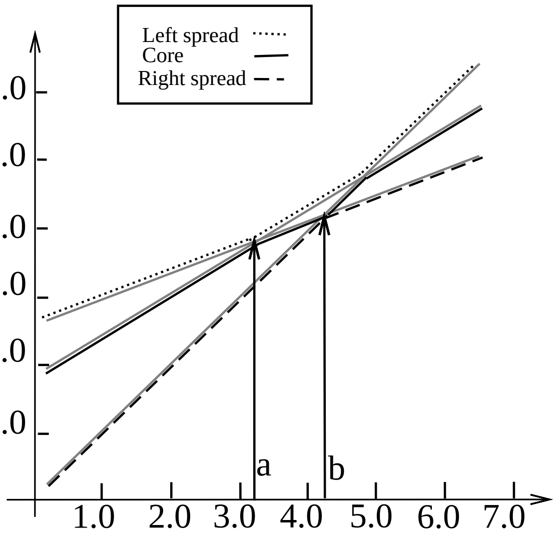


Figure 6.2: An example of a semi-linear fuzzy regression model (taken from [9])

Noting that the input variables are fuzzy triangular numbers, we have $(x_i^l, x_i^c, x_i^r) = (x_i^1, x_i^2, x_i^3)$. Thus, the 3-tuple given in (6.8) can be converted to a fuzzy triangular number after applying a sort operation as in (6.9).

$$\begin{aligned}\widetilde{y} =&(y^l, y^c, y^r) = sort(y^1, y^2, y^3) \\ =&sort(a_n^1 x_n^l + a_{n-1}^1 x_{n-1}^l + \cdots + a_1^1 x_1^l + a_0^1, \\ &a_n^2 x_n^c + a_{n-1}^2 x_{n-1}^c + \cdots + a_1^2 x_1^c + a_0^2, \\ &a_n^3 x_n^r + a_{n-1}^3 x_{n-1}^r + \cdots + a_1^3 x_1^r + a_0^3) \\ =&sort(\sum_{i=1}^{n} a_i^1 x_i^l + a_0^1, \sum_{i=1}^{n} a_i^2 x_i^c + a_0^2, \sum_{i=1}^{n} a_i^3 x_i^r + a_0^3)\end{aligned} \tag{6.9}$$

Data set	SIP-free Model
Dataset (given Table 5.1)	$\widetilde{y} = Sort((1.10, 1.25, 1.42)x + (5.10, 6.76, 8.31))$
Dataset (given Table 5.13)	$\widetilde{y} = Sort((1.22, 1.24, 1.33)x + (-5.82, -5.83, -5.88))$
Dataset (given Table 5.16)	$\widetilde{y} = Sort((0.72, 0.72, 0.72)\widetilde{x} + (0.83, 0.46, 0.09))$
Dataset (given Table 5.19)	$\widetilde{y} = Sort((0.93, 0.12, 0.91)x + (13.10, 7.86, 6.68))$

Table 6.5: Sought SIP-free models for different datasets in the literature

In a simplified version where there is only one input variable, the model in (6.9) can be represented as $\widetilde{y} = sort(a_1^1 x_1^l + a_0^1, a_1^2 x_1^c + a_0^2, a_1^3 x_1^r + a_0^3)$.

Other fuzzy numbers (such as bell shape and etc.) can be treated similarly by application of the level sets representation.

6.2.4 Experimental results

The model given in (6.9) with triangular fuzzy numbers can be represented by $3n+3$ parameters. In case of having a symmetric model the number of the parameters decreases to $2n+2$ which is less than the parameters in models proposed by Lu and Wang [86] and Chen and Dang [87] and equal to the ones proposed by Modaress et al. [135] and Nasrabadi and Nasrabadi [136].

The proposed SIP-free approach is applied on different data sets; crisp input-fuzzy input as given in Tables 5.1, 5.7, 5.11, 5.13, 5.19, and fuzzy input-fuzzy output as given in Table 5.16, the results are compared to the ones obtained by the proposed FLR approach in Chapter 4. The sought models with their errors are reported in Tables 6.5, 6.6. The results reveal that the models obtained from the proposed approach are either superior or Pareto-equivalent to the models defined in Chapter 4.

	NIA		$RNIA$		Hojati's criterion	
Data Set	SIP-free model	FLR model	SIP-free model	FLR model	SIP-free model	FLR model
Dataset (given in Table 5.1)	8.3986	8.3678	3.4961	3.5438	0.4522	0.494
Dataset (given in Table 5.13)	6.8716	6.89	8.4455	10.29	0.7522	0.7056
Dataset (given in Table 5.16)	5.0063	5.265	8.0206	8.3046	0.9513	0.7508
Dataset (given in Table 5.19)	3.6803	5.20	1.973	-	0.2841	0.3364

Table 6.6: Comparison of the SIP-free errors with FLR model on different datasets in the literature

6.3 Summary

A two-stage approach is proposed; first to detect the outliers in a data set and second to fit a model on the clean dataset. The chosen model fitting measure value is minimized with a hybrid optimization method. The benefits of this approach is that there is no need to set any user defined variables and no need to assume the number of outliers from the beginning. Also, another advantage of using meta-heuristics for minimizing the model fitting measurement is that any kind of measurement can be used regardless of being continuous or differentiable. So, the approach is flexible with selection/designing of measurements based on the clients' requirements and requests or the nature of the dataset.

Spread increasing problem (SIP) is also introduced as one of the major issues in fuzzy linear regression. A SIP-free fuzzy linear regression model is formulated to tackle the problem. For doing so, two algebraic operations are introduced and then the model is derived by a hybrid optimization method. The results shows that the obtained SIP-free models are either Pareto-equivalent to or better than existing methods based on four similarity measures.

7

Real-world applications of global continuous optimization and fuzzy linear regression[1]

This chapter discusses the application of the global continuous optimization method and fuzzy linear regression analysis for the design of a pressure vessel and the quality of service for web services.

7.1 Real-world application: The design of a pressure vessel

An optimized design of a cylindrical pressure with two hemispherical heads is sought such that it minimizes the total cost. This example has been used by many

[1] This chapter is based on a journal paper; [2] co-authored by M. A. Orgun and W. Pedrycz. and a conference paper; [10] co-authored by L. Li, M. A. Orgun, and Y. Wang

other studies in the literature to demonstrate the potential application of the proposed approaches [69, 149–153]. The total cost includes the cost of material, forming and welding processes. It is reported that the total cost can be effected by four variables given in Figure 7.1; thickness of the shell x_1, thickness of the head x_2, the inner radius of the vessel x_3 and the length of the cylinderical section of the vessel x_4 [152]. The thickness for the shell and the head can be just multiples of 0.0625, which is the plates thickness. The domain for the length and the radius is continuous. The minimization of the total cost with its constraints is formulated as follows:

$$\min f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

subject to

$$\begin{aligned}
&g_1(x) = -x_1 + 0.0193x_3 \leq 0\\
&g_2(x) = -x_2 + 0.00954x_3 \leq 0\\
&g_3(x) = -\pi x_3^2x_4 - \tfrac{4}{3}\pi x_3^3 + 1296000 \leq 0\\
&g_4(x) = x_4 - 240 \leq 0\\
&0 \leq x_1, x_2 \leq 100\\
&10 \leq x_3, x_4 \leq 200
\end{aligned}$$

The above optimization problem has been studied in [69, 149–153]. The result by our method is compared to the ones reported in the literature, see Table 7.1.

To satisfy the thickness condition, the algorithm is allowed to generate any values for x_1 and x_2 and then in the objective function, before calculating the total cost, they are rounded to the closest number, which is a multiple of 0.0625. Moreover to avoid solutions which do not satisfy g_1, g_2, g_3 and g_4, a *constraint penalty scheme* is applied. In the constraint penalty scheme, instead of avoiding generating solutions vector which do not satisfy any of the constraints, any solution vector is generated and then evaluated but for each of conditions that is not satisfied, a very large value is added to the objective function. For each condition which is not satisfied the objective function is increased by with 100,000.

Table 7.1: Comparison of the best solution for the pressure design vessel by different methods (taken from [2])

Method	Total cost	x_1	x_2	x_3	x_4	Ave. number of func. eval.	Conditions are satisfied?
This work	6059.7153	0.8125	0.4375	42.10	176.64	944	yes
He and Wang [151] (2007)	6061.0777	0.8125	0.4375	42.09	176.75	200,000	yes
Coelho (2008) [149] (2008)	6112.5619	0.8750	0.4375	45.28	141.41	1000	yes
Coello [150] (2000)	6288.7445	0.8125	0.4375	40.32	200.00	900,000	yes
Kannan and Kramer [152] (1994)	7198.0428	1.1250	0.6250	47.70	117.70	not reported	yes
Mahdavi et al. [153] (2007)	5849.7617	0.750	0.375	38.86	221.37	not reported	no
Zahara and Kao [69] (2009)	5930.3037	0.8036	0.3972	41.64	182.41	80,000	no

He and Wang [151] reported to have superior performace over Coello's method [150] both in minimizing the cost and number of evaluations. He and Wang [151] reported the function evaluations for theirs and Coello's method to be 200,000 and 900,000, respectively. As mentioned earlier, one of the aims in GCO is to reduce the number of function evaluations, while minimizing the average error and maximizing success rate. Here, the success rate cannot be computed, as the global minimum is unknown. Our method finds the smallest total cost in comparison with the methods that satisfy all the conditions. Moreover the number of the function evaluations is much smaller than the ones reported by He and Wang [151], and Coello's [150].

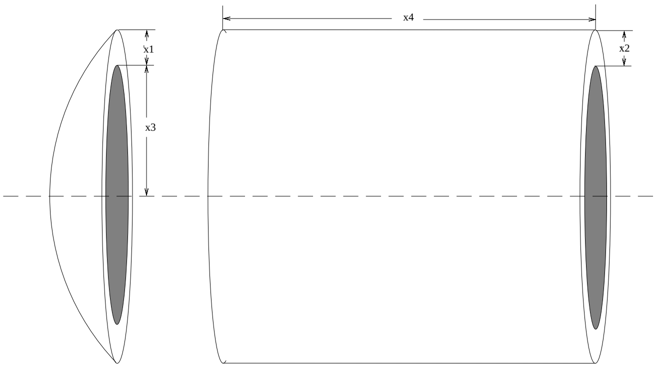


Figure 7.1: The structure of the pressure design with its effective variables on total cost (taken from [2])

The minimized cost for the vessel pressure design is obtained by the approach of Mahdavi et al. [153]. However, the problem with their solution is that it is not acceptable as $x_4 = 221.37$ and it is not in the x_4's domain which is given as $[10, 200]$. The optimized solution by the approach of Zahara and Kao [69] is neither acceptable as it does not satisfy the thickness condition which says both x_1 and x_2 should be a multiple of 0.0625.

The above optimization problem is just an example of how to apply the method for any real-world optimization problems. In particular, the problems which are very hard to be solved by gradient descent-based methods, or the ones which have computationally expensive objective functions are very good candidates to be solved by the proposed methods.

7.2 Application of fuzzy regression to prediction of quality of service

With the advent of service-oriented computing, the issue of trust and Quality of Service (QoS) have become increasingly important. In service-oriented environments, when there are a few service providers providing the same service, a service client would be keen to know the trustworthiness of each service provider in the forthcoming transaction. The trust rating of a delivered service from a service provider can be predicted according to a set of advertised QoS data collected by the trust management authority. Although trust and QoS are qualitative by nature, most data sets represent trust and QoS in the ordinal form for the sake of simplicity.

A new approach based on Fuzzy Linear Regression Analysis (FLRA) is inroduced to extract qualitative information from quantitative data and so use the obtained qualitative information for better modeling of the data. For verification purposes, the proposed approach can be applied for the trust prediction in the forthcoming transaction based on a set of advertised QoS in service-oriented environments.

7.2.1 Why fuzzy linear regression for quality of service prediction

In Service-Oriented Computing (SOC), a service is an autonomous and platform-independent computational entity, which can be described, published, discovered and dynamically assembled for developing massively distributed systems. In fact, any piece of code or application component deployed on a system can be taken as a service [154].

In SOC environments, QoS (Quality of Service) refers to various nonfunctional characteristics. These specified characteristics should be measurable and constitute a description of what a service can offer. The QoS of an IT service is often expressed in terms of capacity, latency, bandwidth, the number of service requests, and the number of incidents [155]. QoS is essential when a set of quality metrics have to be achieved during service provision [155]. In contrast, in SOC environments, the general quality of a delivered service can be represented by a trust rating given by a service client. As trust and QoS both focus on the quality of service, the trust rating of a service or service provider can be taken as a function of a set of advertised QoS values reflecting the quality of the service or the service provider, and the trust rating can be predicted by the trust management authority.

When a service client looks for a service from a large set of services offered by different service providers, in addition to functionality and QoS, reputation-based trust is also a key factor for service selection. It is also critical for the trust management authority to be responsible for maintaining the list of reputable and trustworthy services and service providers with their advertised QoS values, and providing them to clients as required [154].

Conceptually, trust is the measure taken by one party on the willingness and ability of another party to act in the interest of the former party in a certain situation [156]. If the trust value is in the range of [0,1], it can be taken as the subjective probability with which, one party expects that another party performs a given action [157].

The trust issue has been widely studied in many applications. In e-commerce environments, a trust management system can provide valuable information to buyers and prevent certain typical attacks [158, 159]. In Peer-to-Peer information-sharing networks, binary ratings work pretty well as a file is either the definitively correct version or not [160]. In SOC environments, an effective trust management system is critical to identify potential risks, provide objective trust results to clients and prevent malicious service providers from easily deceiving clients [161]. Different from P2P information-sharing networks or the eBay reputation management system, where a binary rating system is used [160], in SOC environments, trust data is usually captured in its ordinal form, such as *5-star*, *gold* and *high-class*, due to the abstraction or even the simplicity of representation. This abstraction makes the trust value more understandable for the end users and service clients, but it is not really suitable for any further inference. An intuitively appealing approach towards trust, which is qualitative by nature, is to not quantize the trust but to take the uncertainty which comes from vagueness into consideration in the level of inference [112]. In contrast, the quantization only takes place for representing the trust to the end users or service clients when it is deemed to become necessary [112].

As mentioned above, the trust rating, representing the subjective belief of a service client on the general quality of the delivered service, is both subjective and qualitative. However, by quantization both subjective and qualitative information in the trust data set would be overlooked. For example, two services with the same trust level of *5-star* may not be necessarily the same for a specific purpose or the requirement of a service client. In fact, one service could be much more suitable than the other one, but, due to the quantization process, such kind of important information may be ignored in the service selection process.

This thesis proposes a QoS based approach to trust prediction using Fuzzy Linear Regression Analysis (FLRA). This approach is built up with the ordinal trust data which can be extracted, so that the hidden fuzzy information can be taken into consideration where it has been overlooked in the original trust data set. The motivations of applying FLRA for predicting trust are summarized as follows.

- Capturing the hidden fuzziness: we want to capture the fuzziness overlooked in the ordinal trust data set due to the reasons such as abstraction. The other machine learning approaches such as neural networks, decision trees and support vector machines in their original form, do not take the fuzziness into consideration and only model the original trust data set.
- Having a transparent model: we want to seek for a transparent relationship between the delivered trust rating and the advertised QoS values where the underlying function is extractable. In industry, neural networks or any other approaches, which have the black-box nature, are not suitable for marketing purposes.
- More detailed information: we want to provide more information than only a few trust levels. For example, a web service is not necessarily either *platinum* or *gold*; it could be for instance *better than gold*, but *worse than platinum* - which can be represented by a fuzzy value.

FLRA is a powerful tool to model the relationships in situations where the attributes are inexact. FLRA is widely applied in economy, finance, marketing, social sciences, healthcare and other domains [101, 106, 162]. In addition, trust is also a widely studied and complex issue, including many uncertain factors [158]. Hence, with the fuzziness, fuzzy regression based on fuzzy set theory can deal with QoS based trust prediction in a reasonable manner. In this thesis, we propose a method for the prediction of trust rating based on fuzzy linear regression for building up a mapping from a set of advertised QoS values to the trust rating of the corresponding delivered service of a service provider. Then, the service client can predict the trustworthiness of the service provider in the forthcoming transaction with its advertised QoS values.

7.2.2 Trust prediction based on fuzzy regression

The application of FLRA for the prediction of trust based is investigated on a set of advertised QoS values. The main goal in this thesis is to predict the trust rating of a web-service, which could be represented by linguistic expressions such as platinum, gold, silver and bronze. So, a common approach is to seek the relationship between the independent attributes (the QoS attributes) and dependent attribute (the trust attribute). For doing so, a numeric form of the trust attribute is considered to construct the model - even if the QoS data comes with some imprecision or uncertainty. Hence, initially the linguistic expressions of trust are all considered to be ordinal data. Then, after they are quantized (defuzzified), classical approaches can be applied [112].

The critical issue here is that the quantization process causes some important information to be overlooked or neglected [6, 89]. In environments with uncertain data, the best option is then to not defuzzify the data but to take the uncertainty in the data into consideration in the level of inference [112]. However, the final results could be defuzzified for the presentation purposes only if it becomes necessary [112].

A linear regression model in the form of

$$Y = a_0 + a_1 x_1 + \cdots + a_n x_n,$$

where n is the number of input variables, is applied for QoS based trust estimation for each web-service. However, we have added an extra term to fuzzify the results. A triangular fuzzy value, like $\widetilde{B_i}$ which derives from

$$(\sum_{i=1}^{n} x_i) \cdot (b_1, b_2, b_3),$$

where $b_1 \leq b_2 \leq b_3$ is added to the above linear regression model given in Eq. (7.1).

$$\widetilde{Y_i} = a_0 + a_1 x_1 + \cdots + a_9 x_9 + \widetilde{B_i} \tag{7.1}$$

The coefficients $a_0, a_1, \cdots, a_n$ are estimated by statistical linear regression. The motivation of applying the model given in Eq. (7.1) is due to the fact that this model does not have the *spread increasing problem* - an issue which exists in the original models given in Eqs (3.9) and (3.10). A model with the spread increasing problem would increase the uncertainty of the trust of a service by an increase in the magnitude of the input QoS variables.

To derive the best values of the variables b_1, b_2 and b_3, tabu-harmony search is applied [6]. In order to measure the goodness of the individual models in tabu-harmony search, we need to count how many predictions of the model are correct. However, the models have a fuzzy estimation of the trust value which is hard to be compared with the actual trust values which are ordinal values. One way to tackle this issue is to defuzzify the estimation and then assign it to a class of trust which has the smallest distance to it. Basically by doing so, the fuzziness is taken into consideration for inference and the quantization process takes place only in the final stage for the sake of presentation [112].

The same margin for all the classes by assigning a class for each estimation is studied. For example, if the margin is 0.2, then in our example estimations will be in the bands of

$$(-\infty, 1.2],\ (1.2, 2.2],\ (2.2, 3.2],\ \text{and } (3.2, +\infty).$$

We introduce the variable b_4 to derive the best margin. Since the value of b_4 depends on the selection of b_1, b_2 and b_3, we optimize all the four variables in one phase.

In contrast to the other methods, with our method, the trust rating will be estimated in terms of a fuzzy number. In fact, trust is naturally an inexact concept. However, due to the abstraction, it is usually converted to a traditional classification problem and the fuzziness is overlooked.

7.2.3 Experiments

The most comprehensive data sets for web-services classification used by [163–165], include ten crisp independent attributes and one linguistic expression which represents the trust level of the web-service. To the best of our knowledge, in most studies in the literature, the trust terms (being platinum, gold, silver and bronze) are quantized to 1 to 4, respectively.

In the experiment, the web-services data set used for the simulation purposes is taken from [15]. The data set contains 365 real web service implementations extracted from the Web. These services were collected using the Web Service Crawler Engine (WSCE). The majority of the Web services were obtained from public sources on the Web including Universal Description, Discovery, and Integration (UDDI) registries, search engines, and service portals.[1] From the list of attributes given in Table 7.2, we have excluded *WSRF* due to its high dominance [163].

[1] For more information visit:
http://www.uoguelph.ca/~qmahmoud/qws/index.html#Service_Classification_

Table 7.2: Web-service dataset; attributes and descriptions (taken from [15])

ID	Attribute name	Description	Units
x_1	Response Time	Time taken to send a request and receive a response	ms
x_2	Availability	Number of successful invocations/total invocations	%
x_3	Throughput	Total Number of invocations for a given period of time	$\frac{invokes}{second}$
x_4	Successability	Number of response / number of request messages	%
x_5	Reliability	Ratio of the number of error messages to total messages	%
x_6	Compliance	The extent to which a documentation follows specification	%
x_7	Best Practices	The extent to which a Web service follows WS-I Basic Profile	%
x_8	Latency	Time taken for the server to process a given request	ms
x_9	Documentation	Measure of documentation (i.e. description tags) in WSDL	%
x_{10}	WSRF	Web Service Relevancy Function	%
y	Service Classification	Levels representing service offering qualities (1 through 4) (trust rating)	Type

The settings of experiments in this section are as follows.

- The introduced data set with the properties given in Table 7.2 is first normalized and then we perform 10-fold cross validation as used in [163].
- Tabu-harmony search with a population size of 100 for 2000 iterations is used.
- The migration and crossover fractions are set to 0.7 and 0.05, respectively.
- The lower and upper bounds of b_1, b_2, b_3 and b_4 are set to -1 and 1, respectively.

The comparison of the method and the results obtained from Support Vector Machines (SVM), decision tree and linear regression are given in Table 7.3. We can observe the following results.

- The accuracy of the proposed method based on fuzzy linear regression is about 80%, which is superior to the other methods compared in this thesis.
- The standard deviation is estimated to be 0.07, which reveals the robustness of the approach on different runs.
- Both the standard deviation and the performance of our proposed approach based on fuzzy linear regression is better than those of the one which applies the traditional linear regression.
- Moreover, as discussed earlier, our approach is able to re-generate the overlooked uncertainty in the trust of service and use it for inference.
- The standard deviations and the training performances of the other methods are not given in Table 7.3, as they are not reported in [163].

Table 7.3: Accuracy of the classifiers after removing WSRF (taken from [10])

Classifier	Average performance on testing sets (%)	Standard deviation of the testing sets errors	Average performance on training sets (%)	Standard deviation of the training sets errors
Fuzzy linear regression (This work)	79.46	0.0756	83.36	0.0060
Decision Trees (CART) [163]	78.61	-	-	-
Linear regression (This work)	74.17	0.0816	79.91	0.0095
SVM [163]	60.55	-	-	-
Decision trees (J48) [163]	67.77	-	-	-

As an example, we have estimated the trust ratings of two web services from the obtained model during training. The trust values are in the form of fuzzy triangular numbers as follows:

$$(1.5714, 3.4027, 4.1017),\ (1.9139, 3.8692, 4.6155).$$

Although after defuzzification both would be classified as *gold*, there are differences between them in terms of QoS. In fact, Figure 7.2 reveals that the second service is slightly better than the first one. Moreover, there is an uncertainty that the quality of both services depicted in Figure 7.2 might actually be silver, as the left spreads of both the fuzzy numbers are very wide.

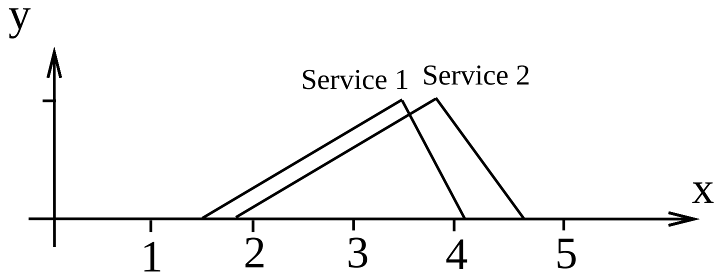


Figure 7.2: The comparison of the trust of two web services before defuzzification (taken from [10])

As another example, we assume the trust of two web services given in Figure 7.3. In case of using traditional (exact) methods, it is deduced that service 2 is superior to service 1. However, as the left spread of service 2 goes all the way to the bronze quality, this may result in selecting service 1 over service 2.

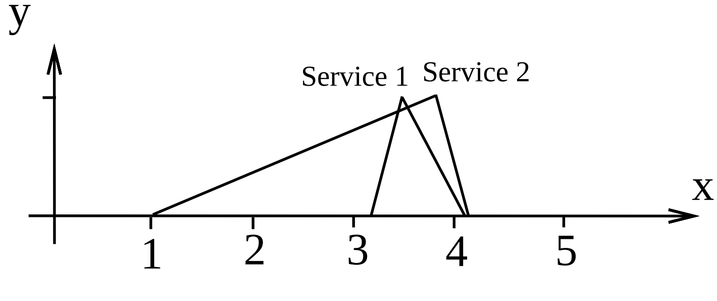


Figure 7.3: An example where a traditional method and an inexact method may vary in the result (taken from [10])

7.3 Summary

The application of global continuous optimization is shown for designing of a pressure vessel. This is an optimization problem with constraints where the goal is to minimize the cost of production for making a pressure vessel. The results are discussed and compared with the other methods in the literature.

An approach based on fuzzy linear regression is applied to extract the qualitative information which may have been overlooked in the original data set. The extracted qualitative data is then used for modeling the data. The proposed approach is applied for the trust prediction of the delivered web-services based on a set of advertised QoS values. The method not only provides more detailed information of trust but it also has promising results. Although in order to quantitatively compare the performance of the method, the results are defuzzified, one could use the fuzzy results as they provide more information.

8 Conclusions and future work

We have investigated global continuous optimization methods based on meta heuristic approaches for optimization problems where the objective function has the following characteristics;

- computationally expensive, and/or
- the differentiability (or even the objective function itself is a black-box) is unknown due to reasons such as being a trade secret.

Fuzzy linear regression analysis can be modeled as an optimization problem where the objective function can be non-transparent and so it is beneficial to apply global continuous optimization. We have shown the application of global continuous optimization for deriving a fuzzy linear model, detecting outliers and also solving the spread increasing problem.

8.1 Conclusions of each chapter

Slow rate of convergence is a drawback in some of meta-heuristic based approaches. Hybridization is a solution toward solving this issue. In Chapter 2, we classified the global continuous optimization methods into three main categories according to their hybridization approach. It is noted that the simple *diver-intens* hybridization framework is very fast when compared to other frameworks, but since only one best solution is considered for the intensification phase, it does not guarantee finding the optimum solution. On the other hand, iterative *divers-intens* hybridization frameworks increase the possibility of finding the global optimum solution. But this framework is computationally very expensive since intensification is carried out for all of the potential areas. *Divers-semiintens-intens* frameworks are realized to keep the balance between the computational cost and the rate of finding global optimum solutions. The results from our proposed global continuous hybrid method, given in Appendix A.1, advise that our approach is either superior or Pareto-equivalent to the other methods in the literature.

In Chapter 3, fuzzy linear regression analysis was modelled as an optimization problem where the goal is to minimize the model fitting measure. Since the fitting measure is not always transparent or simple to be evaluated, global continuous optimization methods are ideal solutions. We have also introduced a protocol for comparing the performance of two models against fitting measures, in which a model could be better, worse or Pareto-equivalent than/to another model on a given data set.

In Chapter 4, we proposed an approach based on tabu and harmony search to tackle fuzzy linear regression analysis. The experimental results show that the models obtained by global continuous optimization is either superior or Pareto-equivalant to the ones in the literature. This is due to the fact that a fitting measure applied in our approach can be used as the objective function while in least square or linear programming based approaches the goal is to minimize a distance function and the performance is then measured via a different objective function.

In Chapter 5 we have investigated the goodness of the proposed fuzzy linear regression model based on global continuous optimization. According to the experimental results the proposed method outperforms or Pareto-equivalent to other methods compared on fuzzy input-fuzzy output and crisp input-fuzzy output data sets. The proposed approach could be generalized for any type of convex fuzzy value. The CPU times reported for deriving the models are more than LS and LP based approaches in the literature, however the models can still be obtained in a feasible time. The CPU time varies based on the size of the data set and type of the objective function, e.g. for Euclidean type of objective function the optimization is much faster compared to an area based objective function such as $RNIA$.

In Chapter 6, outlier detection and spread increasing problems are also identified as problems in fuzzy linear regression which can be tackled by global continuous optimization. We have proposed a two stage fuzzy linear regression analysis where in the first stage the objective is to exclude the data which do not follow the trend of the rest of the data set. The second stage is dedicated to fit a model with the minimum error on the clean data set. The experimental results provide evidence that the proposed approach has promising features such as being practical in terms of removal and fitting a model, and not requiring to configure control variables such as the ones in other models. The spread increasing problem on the other hand is tackled by introducing an objective function which applies new algebraic operations. The sought model is capable of having variable spreads and is shown to be superior or Pareto-equivalant to other models in the literature.

In Chapter 7, we have studied two real-world applications for global continuous optimization and fuzzy linear regression. In the first part, an hybridized approach is applied to reduce the cost of manufacturing of a pressure vessel. This is an example where the accuracy of result is more of an interest compared to optimization speed. This is due to the fact that the design process is an off-line procedure. We have also investigated an example with regards to the prediction of the quality of web-services by fuzzy linear regression. We have shown that fuzzy linear regression analysis could be applied to capture qualitative information from quantitative data. This example shows a methodolgy of applying fuzzy linear regression analysis for the applications where there is uncertainty naturally but due to reasons such abstraction, loss of information and etc the uncertainty is not recorded. The results show that not only fuzzy linear regression has promising performance but it also extracts more information in terms of the uncertainty which exists in the quality term.

8.2 Future research direction

There remain many avenues for future work and we briefly suggest some of them below;

- The data sets used in the majority of the fuzzy regression literature are either produced synthetically or are originally crisp data sets which are converted to fuzzy data set. It would be very beneficial to have a real fuzzy data set were the input and output data are fuzzy values by nature. An ultimate way of collecting fuzzy data could involve running focus group studies. Since the uncertainty should come from the participants rather than having another person fuzzifying the data, the participants have to be trained how to put their feedback in terms of fuzzy values.

- It would be useful to investigate muti-objective optimization methods for the case that there are more than one model fitting measures to be maximized. In cases where the fitting measures may come from different clients and all has to be maximized, multi-objective global continuous optimization methods based on meta-heuristic approaches seem to be ideal solutions. The application of multi-objective optimization method seems to decrease the rate of Pareto-equivalant models compared to the ones in the literature and increase the superiority rate of the models.

- The performance of fuzzy regression models are usually measured based on the same data to which the model is fitted. A more practical way of measuring the performance of a model could be to measure how good the model predicts the test data. As otherwise a model may perform very well on the training set but poor on new data. A future study may consider introducing a protocol for measuring the performance of a model based on a testing data set.

- It would be also beneficial to compare the results obtained from fuzzy linear regression with the ones obtained from conventional models such as statistical linear regression. According to the results obtained in Chapter 7, it seems that for the problems where uncertainty naturally exists but overlooked for any reasons, fuzzy linear regression may have better performance. A comprehensive study could be conducted to investigate this speculation.

- It would be beneficial to investigate the advantage and disadvantage of different similarity measures in the literature. We have suggested some benefits of few similarity measures, however a comprehensive investigation dedicated to study the application of each similarity measure would be useful.

- Application of other soft computing approaches such as genetic programming, K Nearest Neighbours (K-NN) also seem to have potential for non-parametric fuzzy regression. Approaches such as genetic programming can be applied for situations where the mathematical formula of a model is required, otherwise approaches such as K-NN seems to be applicable too.

- Crisp data sets with many entries could be abstracted by clustering the data and representing them by fuzzy values. It would be interesting to compare and contrast a model derived from fuzzy linear regression which fits the abstracted fuzzy data with a model derived from applying a statistical linear regression on crisp data. In a particular case when a data set is very noisy, the fuzzy model might not be as sensitive as the crisp model to the noise.

A

Appendices[1]

A.1 Optimization results for benchmark functions

To show the effectiveness of the proposed method three important parameters: convergence, speed and robustness may be studied [22]. Each of these parameters is associated with the average error (Av_e), the number of function evaluations (f_e) and the success rate (S_r), respectively. Firstly, we report the experimental results for the benchmark functions and secondly a real world application from the literature is solved by the method.

Robustness states that the method can be applied to find the global minimum solution of different functions. For the sake of robustness 17 multimodal benchmark functions (2-10 variables) known in the literature are considered (as listed in Appendix A) [22, 24, 26–28, 49, 51, 61–64, 66, 67, 69, 80]. Since these are multimodal functions, it is hard to find their global minimum [27]. To avoid immature conclusions due to the choice of a random initial population in the partitioning phase and applying randomization in the neighbour search-strategy, the method was run for 100 times for each of the functions. In each run, different initial seeds are applied

[1] This chapter is based on journal papers; [6] co-authored by M. A. Orgun, M. Mashinchi and W. Pedrycz, and [2] co-authored by M. A. Orgun and W. Pedrycz.

to generate random initial solutions in the partitioning phase. Thus each run starts from different starting points. The success rate is defined as $\frac{g}{100}$ where g is the number of successfull runs. A run is considered successful if it satisfies the following condition [16,24,26,28,51,62,64]:

$$|f^* - f| < \epsilon_1.|f^*| + \epsilon_2,$$

where f^* is the known global minimum of the function under discussion and f is the global minimum of that function which our method computes. The coeffcients ϵ_1 and ϵ_2 are set to 10^{-4} and 10^{-6} respectively, see also [16,24,26–28,51,61,62,80].

To test the convergence rate, the relative distance between the obtained global minimum solution and the actual one is computed. The average distance in 100 runs returns the average error.

The time which takes for the method to find the global minimum of a solution determines the speed of the method. However, since computing the optimization time strictly depends on the computer's speed [22], in the literature the speed criterion is determined by the average number of function evaluations in all runs [22,24,26–28,49,51,55,61–64,66,67,69,80].

Note that it is unfair to use the global optimum solution value as supportive information for the stopping condition. So, we do not use the analytical global optimum solution of the function till the end where we just want to compute the success rate and the average error.

One of the factors that has a considerable effect on the result is the sizes of the STTL and LTTL. We examine the effect of different STTL and LTTL sizes on the solution quality and computational effort for acquiring the minimum solution. The experimental results of three different combinations of the STTL's size and LTTL's size ((2,2),(5,5) and (10,10)) are reported in Table A.1. we pursue the iteration until after K consecutive iterations where the improvement is less than ϵ as given below:

$$|f(x_k) - f(x_{k-n})| < \epsilon, \tag{A.1}$$

where ϵ is used to control the termination of the method. The ϵ in (A.2) is set to 10^{-10} for all of 100 runs for each of the functions. The simulations were run on Pentium IV 3.00GHz with 2GB memory. The program is coded in Matlab 7.0. According to the result from Table A.1, as expected the different TL sizes have effect on the success rate of the algorithm. The selection of the large TL size makes the algorithm to push the next solutions in new areas, so it gives more opportunities to explore a broader search space. The sizes of both STTL and LTTL is set to 10 in all subsequent comparisons with other methods.

$$|f(x_k) - f(x_{k-n})| < \epsilon, \tag{A.2}$$

where ϵ is used to control the termination of the method.

Table A.1: Effect of STTL and LTTL sizes on the convergence, speed and robustness of the proposed method (taken from [2])

Benchmark function	Performance	TL size		
		(2,2)	(5,5)	(10,10)
RC	$S_r(\%)$	100	100	100
	Av_e	$5.3588e-11$	$5.1376e-11$	$5.4827e-11$
	f_e	134	146	146
	$Av_t(s)$	0.0171	0.0179	0.0171
GP	$S_r(\%)$	100	100	100
	Av_e	$5.5194e-11$	$5.6707e-11$	$5.3463e-11$
	f_e	165	166	165
	$Av_t(s)$	0.0222	0.0209	0.0190
R_2	$S_r(\%)$	100	100	100
	Av_e	$5.6121e-11$	$5.1500e-11$	$5.4984e-11$
	f_e	223	221	222
	$Av_t(s)$	0.0215	0.0179	0.0222
Z_2	$S_r(\%)$	100	100	100
	Av_e	$5.1835e-11$	$6.0981e-11$	$5.4537e-11$
	f_e	138	139	139
	$Av_t(s)$	0.0155	0.0153	0.0153
$H_{3,4}$	$S_r(\%)$	100	100	100

Continued on next page

Table A.1 – continued from previous page

Benchmark function	**Performance**	**TL size**		
		$(2,2)$	$(5,5)$	$(10,10)$
	Av_e	$5.7063e-11$	$6.1536e-11$	$5.5487e-11$
	f_e	207	212	212
	$Av_t(s)$	0.0275	0.0288	0.0264
DJ	$S_r(\%)$	100	100	100
	Av_e	$5.4476e-11$	$5.4229e-11$	$5.6959e-11$
	f_e	199	199	199
	$Av_t(s)$	0.0223	0.0225	0.0225
SH	$S_r(\%)$	86	93	100
	Av_e	$5.4655e-11$	$5.6929e-11$	$5.3745e-11$
	f_e	358	383	459
	$Av_t(s)$	0.0391	0.0426	0.0510
B_2	$S_r(\%)$	80	82	83
	Av_e	$5.5337e-11$	$5.4287e-11$	$5.5828e-11$
	f_e	560	584	658
	$Av_t(s)$	0.0655	0.0733	0.0760
ES	$S_r(\%)$	76	77	78
	Av_e	$8.2305e-11$	$2.2066e-11$	$4.4793e-11$
	f_e	510	521	515
	$Av_t(s)$	0.0576	0.0617	0.0640
$Shl_{4,5}$	$S_r(\%)$	73	72	74
	Av_e	$6.2260e-11$	$6.2174e-11$	$6.9967e-11$
	f_e	799	837	840
	$Av_t(s)$	0.0957	0.1003	0.1057
$Shl_{4,7}$	$S_r(\%)$	83	87	88
	Av_e	$6.3011e-11$	$6.2403e-11$	$6.4606e-11$
	f_e	801	841	845

Continued on next page

Table A.1 – continued from previous page

Benchmark function	Performance	TL size		
		$(2,2)$	$(5,5)$	$(10,10)$
	$Av_t(s)$	0.0961	0.1026	0.1052
$Shl_{4,10}$	$S_r(\%)$	84	87	89
	Av_e	$6.1137e-11$	$6.5839e-11$	$5.9734e-11$
	f_e	800	843	845
	$Av_t(s)$	0.0962	0.1012	0.1049
$H_{6,4}$	$S_r(\%)$	66	64	66
	Av_e	$8.6781e-11$	$8.1020e-11$	$8.0286e-11$
	f_e	873	878	880
	$Av_t(s)$	0.1129	0.1150	0.1190
R_5	$S_r(\%)$	94	94	95
	Av_e	$7.1966e-11$	$7.0860e-11$	$7.1356e-11$
	f_e	1038	1040	1033
	$Av_t(s)$	0.1229	0.1205	0.1236
Z_5	$S_r(\%)$	100	100	100
	Av_e	$7.2742e-11$	$6.9021e-11$	$7.1166e-11$
	f_e	674	674	674
	$Av_t(s)$	0.0759	0.0815	0.0853
R_{10}	$S_r(\%)$	85	85	87
	Av_e	$1.8648e-10$	$4.1410e-10$	$8.9220e-10$
	f_e	4039	4051	4057
	$Av_t(s)$	0.7093	0.7563	0.7525
Z_{10}	$S_r(\%)$	100	100	100
	Av_e	$1.9313e-10$	$1.7252e-10$	$1.4605e-10$
	f_e	1899	1899	1898
	$Av_t(s)$	0.3086	0.3171	0.3184

To show the efficiency of the proposed method, it is compared to the best and the most recent other published hybrid methods given in Table A.2. For comparison, we say a $GCOM_1$ is better than a $GCOM_2$ for a function, only if all of the following "*GCOM comparison conditions*" simultaneously hold true:

1. f_e of $GCOM_1 < f_e$ of $GCOM_2$,
2. S_r of $GCOM_1 < S_r$ of $GCOM_2$,
3. Av_e of $GCOM_1 < Av_e$ of $GCOM_2$.

In case for a function only one or two of the conditions hold true, it can be said that the $GCOM_1$ is "*Pareto equivalent*" to the $GCOM_2$ for that function.

To compare our method to other methods, see Table A.2, we need to have smaller Av_e according to the *GCOM comparison conditions*. Thus we set the ϵ value in the Equation (A.2) equal to a litte bit less than the Av_e of others methods . So if for example a GCOM for a function returned $Av_e = 10^{-12}$ then we set ϵ in our method equal to 10^{-13} to make sure that our method can obtain a smaller Av_e. Note that this does not mean that we assume that we know the analytical global minimum solution of the functions. The value of each control variable that we use in our experiments is given in Table A.3.

Table A.2: List of various GCOMs used for the comparison (taken from [2])

Method	Reference	Published year
Improved TSNM	This work	
GAPSO	[67]	2008
PSACO	[27]	2007
HCIACO	[80]	2007
NMPSO	[76]	2007
DTS	[16, 26]	2004, 2006
CTSS	[24]	2005
NHGA	[166]	2005
CHA	[51]	2003
ETS	[62]	2000

According to the experiments, we can divide the benchmark functions into 3 classes. The first class is that of the low dimensional functions (less than 4 input variables) which have relatively small or medium input domains. The comparison experiments for this class is given in Table A.4. The next class includes the functions which have low dimensions (less than 4 input variables) but have large input

Table A.3: The value of control variables (taken from [2])

Abbreviation	**Description**
$S = 10$	STTl size
$L = 10$	LTTL size
$N = 5$ (if input dimension ≤ 5)	Number of neighbours
$N = 10$ (if input dimension > 5)	
$DF = \frac{upperbound - lowerbound \times \frac{\max(5 - inputdimension, 1)}{2}}{6}$	Distribution factor
$SF_{STTL} = \frac{DF}{100}$	STTL Similarity factor
$SF_{LTTL} = 10 \times SF_{STTL}$	LTTL Similarity Factor
$\alpha, \beta = 0.2$	Step and Directed percentages

domains, Table A.1. The next investigated class compares the results of our method with others for the functions which have relatively high dimensions (more than 3 input variables) regardless their input domains, Table A.1.

Table A.4: Comparison between the proposed method and other GCOMs for low dimensional benchmark functions ($n < 4$) with small or medium input domains ($|a_i - b_i| < 20,\ i = 1, \cdots, n$) (taken from [2])

GCOM	RC			GP			R_2			Z_2		
	f_e	S_r	Av_e	f_e	S_r	Av_e	f_e	S_r	Av_e	f_e	S_r	Av_e
Proposed	**178**	**100**	**6.12e-14**	**165**	**100**	**5.34e-11**	222	**100**	**5.49e-11**	**92**	**100**	**4.93e-5**
GAPSO [67]	8254	100	$9e-5$	25706	100	$1.2e-4$	140894	100	$6.4e-4$	95	100	$5e-5$
Proposed	**178**	**100**	**6.12e-14**	**165**	**100**	**5.34e-11**	222	**100**	**5.49e-11**	247	100	$5.54e-27$
PSACO [27]	209	100	$2.61e-13$	240	100	$0.0e-7$	327	100	$1.71e-10$	**167**	**100**	**5.70e-27**
Proposed	-	-	-	**165**	**100**	**5.34e-11**	222	**100**	**5.49e-11**	-	-	-
HCIACO [80]	-	-	-	34533	100	0.0	18747	100	$1e-8$	-	-	-
Proposed	-	-	-	-	-	-	-	-	-	-	-	-
NMPSO [76]	-	-	-	-	-	-	-	-	-	-	-	-
Proposed	**178**	**100**	**6.12e-14**	**165**	**100**	**5.34e-11**	222	**100**	**5.49e-11**	**161**	**100**	**6.24e-10**
DTS [16, 26]	212	100	$4e-7$	230	100	$5e-9$	254	100	$5e-9$	201	100	$5e-9$
Proposed	**122**	**100**	**3.57e-7**	**114**	**100**	**4.30e-4**	222	**100**	**5.49e-11**	139	100	$5.45e-11$
CTSS [24]	125	100	$5e-3$	119	100	$1e-3$	369	100	$4e-3$	78	100	$3e-7$
Proposed	-	-	-	-	-	-	**226**	**100**	**5.49e-11**	**139**	**100**	**5.45e-11**
NHGA [166]	-	-	-	-	-	-	239	100	$7e-4$	390	100	$3e-6$
Proposed	**178**	**100**	**6.12e-14**	**165**	**100**	**5.34e-11**	222	**100**	**5.49e-11**	**139**	**100**	**5.45e-11**
CHA [51]	295	100	$1e-4$	259	100	$1e-3$	459	100	$4e-3$	215	100	$3e-6$
Proposed	**178**	**100**	**6.12e-14**	**165**	**100**	**5.34e-11**	222	**100**	**5.49e-11**	**139**	**100**	**5.45e-11**
ETS [62]	245	100	$5e-2$	231	100	$2e-3$	480	100	$2e-2$	195	100	$2e-7$

Continued...

Table A.5: (Cont.) Comparison between the proposed method and other GCOMs for low dimensional benchmark functions ($n < 4$) with small or medium input domains ($|a_i - b_i| < 20,\ i = 1, \cdots, n$) (taken from [2])

	$H_{3,4}$			DJ			SH		
	f_e	S_r	Av_e	f_e	S_r	Av_e	f_e	S_r	Av_e
Proposed	**212**	**100**	**5.54e-11**	**199**	**100**	**5.69e-11**	**459**	**100**	**5.37e-11**
GAPSO [67]	2117	100	$2.0e-4$	206	100	$4e-5$	96211	100	$7e-5$
Proposed	**219**	**100**	**4.97e-12**	407	100	$5.86e-29$	**459**	**100**	**5.37e-11**
PSACO [27]	592	100	$2.07e-11$	**190**	**100**	**7.69e-29**	534	100	$1.02e-9$
Proposed	-	-	-	-	-	-	-	-	-
HCIACO [80]	-	-	-	-	-	-	-	-	-
Proposed	-	-	-	**226**	**100**	**5.75e-14**	-	-	-
NMPSO [76]	-	-	-	1957	100	$1.63e-13$	-	-	-
Proposed	**212**	**100**	**5.54e-11**	**199**	**100**	**5.69e-14**	**402**	**100**	**7.21e-6**
DTS [16, 26]	438	100	$2e-6$	446	100	$4e-9$	274	92	$9e-6$
Proposed	**212**	**100**	**5.54e-11**	**149**	**100**	**5.87e-5**	343	100	$6.9e-4$
CTSS [24]	225	100	$5e-3$	155	100	$2e-4$	**283**	**100**	**1e-3**
Proposed	-	-	-	-	-	-	-	-	-
NHGA [166]	-	-	-	-	-	-	-	-	-
Proposed	**212**	**100**	**5.54e-11**	**199**	**100**	**5.69e-11**	**343**	**100**	**6.9e-4**
CHA [51]	492	100	$5e-3$	371	100	$2e-4$	345	100	$5e-3$
Proposed	**212**	**100**	**5.54e-12**	**199**	**100**	**5.69e-14**	**343**	**100**	**6.9e-4**
ETS [62]	548	100	$9e-2$	338	100	$3e-8$	370	100	$1e-3$

Table A.6: (Cont.) Comparison between the proposed method and other GCOMs for low dimensional benchmark functions ($n < 4$)with large input domains ($|a_i - b_i| \geq 20,\ i = 1, \cdots, n$) (taken from [2])

GCOM		B_2			ES	
	f_e	S_r	Av_e	f_e	S_r	Av_e
Proposed	658	83.4	$5.58e-11$	455	78.1	$7.8e-7$
GAPSO [67]	**174**	**100**	**1e-5**	809	100	$3e-5$
Proposed	699	83.4	$4.83-17$	455	78.1	$7.8e-7$
PSACO [27]	**370**	**100**	**5.55e-17**	**254**	**100**	**0.0e-7**
Proposed	-	-	-	-	-	-
HCIACO [80]	-	-	-	-	-	-
Proposed	658	83.4	$5.58e-11$	-	-	-
NMPSO [76]	1124	100	$3.23e-10$	-	-	-
Proposed	-	-	-	455	75.4	$7.8e-7$
DTS [16, 26]	-	-	-	**223**	**82**	**4e-7**
Proposed	581	83.4	$2.07e-8$	450	78.1	$5.68e-4$
CTSS [24]	98	100	$5e-6$	325	100	$5e-3$
Proposed	-	-	-	-	-	-
NHGA [166]	-	-	-	-	-	-
Proposed	581	83.4	$2.07e-8$	450	78.1	$5.68e-4$
CHA [51]	**132**	**100**	**2e-7**	952	100	$1e-3$
Proposed	-	-	-	450	78.1	$5.68e-4$
ETS [62]	-	-	-	1284	100	$1e-2$

Table A.7: Comparison between the proposed method and other GCOMs for high dimensional benchmark input domains ($n \geq 4$) (taken from [2])

GCOM	$Shl_{4,5}$			$Shl_{4,7}$			$Shl_{4,10}$			$H_{6,4}$		
	f_e	S_r	Av_e	f_e	S_r	Av_e	f_e	S_r	Av_e	f_e	S_r	Av_e
Proposed	840	74.9	$6.99e-11$	751	88.8	$5e-6$	751	89.1	$5e-6$	880	66.3	$8.02e-11$
GAPSO [67]	529344	100	$1.4e-4$	56825	100	$1.5e-4$	43314	100	$1.2e-4$	12568	100	$2.4e-4$
Proposed	840	74.9	$6.99e-11$	836	88.8	$6.46e-11$	845	89.1	$5.97e-11$	880	66.3	$8.02e-11$
PSACO [27]	**482**	**100**	**5.82e-11**	**483**	**100**	**1.81e-10**	**489**	**100**	**3.07e-10**	**529**	**95**	**4.47e-11**
Proposed	-	-	-	-	-	-	-	-	-	-	-	-
HCIACO [80]	-	-	-	-	-	-	-	-	-	-	-	-
Proposed	-	-	-	-	-	-	-	-	-	-	-	-
NMPSO [76]	-	-	-	-	-	-	-	-	-	-	-	-
Proposed	**777**	**74.9**	**5e-7**	**751**	**88.8**	**5e-6**	**751**	**89.1**	**5e-6**	880	66.3	$8.02e-11$
DTS [16, 26]	819	75	$3e-7$	812	65	$4e-5$	828	52	$1e-5$	1787	83	$2e-6$
Proposed	691	74.9	$5e-3$	697	88.8	$5e-3$	700	89.1	$5e-3$	-	-	-
CTSS [24]	538	75	$7e-3$	590	77	$1e-3$	555	74	$1e-3$	-	-	-
Proposed	-	-	-	-	-	-	-	-	-	-		
NHGA [166]	-	-	-	-	-	-	-	-	-	-	-	-
Proposed	**691**	**84.9**	**5e-3**	697	88.8	$5e-3$	700	89.1	$5e-3$	880	66.3	$8.02e-11$
CHA [51]	698	85	$9e-3$	620	85	$1e-2$	635	85	$1.5e-2$	930	100	$8e-3$
Proposed	**691**	**74.9**	**5e-3**	**697**	**88.8**	**5e-3**	**700**	**89.1**	**5e-3**	880	66.3	$8.02e-11$
ETS [62]	825	75	$1e-2$	910	80	$1e-2$	898	75	$1e-2$	1520	100	$5e-2$

Continued...

Table A.8: (Cont.) Comparison between the proposed method and other GCOMs for high dimensional benchmark input domains ($n \geq 4$) (taken from [2])

	R_5			Z_5			R_{10}			Z_{10}		
	f_e	S_r	Av_e	f_e	S_r	Av_e	f_e	S_r	Av_e	f_e	S_r	Av_e
Proposed	1033	95.8	$7.13e-11$	**520**	**100**	**6.82e-5**	3533	87	$4.21e-5$	1342	100	$4.39e-6$
GAPSO [67]	1358064	100	$1.3e-4$	398	100	$0.0e-5$	5319160	100	$5e-4$	872	100	$0.0e-5$
Proposed	1033	95.8	$7.13e-11$	811	100	$6.78e-17$	3533	87	$4.21e-5$	**1898**	**100**	**1.46e-10**
PSACO [27]	**517**	**100**	**1.85e-4**	**516**	**100**	**3.63e-17**	**1541**	**95**	**4e-4**	2299	100	$2e-8$
Proposed	-	-	-	-	-	-	-	-	-	-	-	-
HCIACO [80]	-	-	-	-	-	-	-	-	-	-	-	-
Proposed	-	-	-	-	-	-	4057	87	$8.92e-10$	-	-	-
NMPSO [76]	-	-	-	-	-	-	28836	100	$3.37e-9$	-	-	-
Proposed	**1033**	**95.8**	**7.13e-11**	**647**	**100**	**7.11e-11**	**4057**	**87**	**8.92e-10**	**1898**	**100**	**1.46e-10**
DTS [16, 26]	1684	85	$6e-9$	1003	100	$7e-9$	9037	85	$2e-8$	4032	100	$2e-8$
Proposed	-	-	-	-	-	-	-	-	-	-	-	-
CTSS [24]	-	-	-	-	-	-	-	-	-	-	-	-
Proposed	1033	95.8	$7.13e-11$	**674**	**100**	**7.11e-11**	4057	87	$8.92e-10$	**1898**	**100**	**1.46e-10**
NHGA [166]	1660	100	$2e-3$	1310	100	$4e-4$	6257	100	$3e-3$	10734	100	$1e-6$
Proposed	1033	95.8	$7.13e-11$	**674**	**100**	**7.11e-11**	4057	87	$8.92e-10$	**1898**	**100**	**1.46e-10**
CHA [51]	3290	100	$1.8e-2$	950	100	$6e-5$	14563	83	$8e-3$	4291	100	$1e-6$
Proposed	1033	95.8	$7.13e-11$	**674**	**100**	**7.11e-11**	4057	87	$8.92e-10$	**1898**	**100**	**1.46e-10**
ETS [62]	2142	100	$8e-2$	2254	100	$4e-6$	15720	85	$2e-2$	4630	100	$2e-7$

The comparison between the GCOMs shows that the proposed method outperforms other GCOMs when the functions under investigation with n input variables on the domains of $[a_i, b_i]$ have low dimensionality (less than 4 input variables) and relatively small or medium input domains $|a_i - b_i| \leq 20$, for $i = 1, \ldots, n$, Table A.4. The method in [67] yields good results for this class only if the optimum solution is equal to zero vector, $X^* = (0, \cdots, 0)$, otherwise this method in terms of the number of function evaluations is very computationally expensive. Since in reality, we do not know whether the solution is zero or not, it may be better to apply a method which has an acceptable performance regardless of the location of the optimum solutions.

In case the number of input variables increases (more than 3) or the domains of input variables become larger ($|a_i - b_i| > 20$ for $i = 1, \ldots, n$) our proposed method cannot outperform other methods. Although it does not outperform the other GCOMs, it is not worse than many of them since according to the *GCOM comparison conditions*, the proposed one is *pareto equivalant* to some others in many cases. This fact reveals that depending on different applications, different GCOMs for high dimensions can be applied.

A.2 Mathematical formulation of the benchmark function

Branin RCOS (RC) **(two variables)**
Definition: $RC(x_1, x_2) = \left(x_2 - (\frac{5}{4\pi^2})x_1^2 + (\frac{5}{\pi})x_1 - 6\right)^2 + 10\left(1 - (\frac{1}{8\pi})\right)\cos(x_1) + 10;$
Search domain: $-5 < x_1 < 10, 0 < x_2 < 15;$
Local minimum: no local minima;
Global minima: $(x_1^*, x_2^*) = (\pi, 12.275), (\pi, 2.275), (9.42478, 2.475), RC(X^*) \simeq 0.39789.$

Goldstein and Price (GP) **(two variables)**
Definition: $GP(x_1, x_2) = \left(1 + (x_1 + x_2 + 1)^2 \times (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right)$
$\times\left(30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right);$
Search domain: $-2 < x_j < 2, j = 1, 2;$
Local minimum: four local minima;
Global minima: $(x_1^*, x_2^*) = (0, -1), GP(X^*) = 3.$

Rosenbrock (R_n) **(n variables)**
Definition: $R_n(X) = \sum_{j=1}^{n-1}\left(100(x_j^2 - x_{j+1})^2 + (x_j - 1)^2)\right);$

Search domain: $-5 < x_j < 10, j = 1, \ldots, n$;
Local minimum: several local minima;
Global minima: $X^* = (1, \ldots, 1), R_n(X^*) = 0$.

Zakharaov (Z_n) **(n variables)**
Definition: $Z_n(X) = \left(\sum_{j=1}^{n} x_j^2\right) + \left(\sum_{j=1}^{n} 0.5 j x_j\right)^2 + \left(\sum_{j=1}^{n} 0.5 x_j\right)^4$;
Search domain: $-5 < x_j < 10, j = 1, \ldots, n$;
Local minimum: several local minima;
Global minima: $X^* = (0, \ldots, 0), Z_n(X^*) = 1$.

Hartmann $(H_{3,4})$ **(three variables)**
Definition: $H_{3,4}(X) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right)$;
Search domain: $0 < x_j < 1, j = 1, 2, 3$;
Local minimum: four local minima;
Global minima: $X^* = (0.11, 0.555, 0.855), H_{3,4}(X^*) \simeq -3.86278$.

De Joeng (DJ) **(three variables)**
Definition: $DJ(X) = x_1^2 + x_2^2 + x_3^2$;
Search domain: $-5.12 < x_j < 5.12, j = 1, 2, 3$;
Local minimum: no local minima;
Global minima: $X^* = (0, 0, 0), DJ(X^*) = 0$.

Shubert (SH) **(two variables)**
Definition: $SH(x_1, x_2) = \left(\sum_{j=1}^{5} j \cos[(j+1)x_1 + j]\right) \times \left(\sum_{j=1}^{5} j \cos[(j+1)x_2 + j]\right)$;
Search domain: $-10 < x_j < 10, j = 1, 2$;
Local minimum: 760 local minima;
Global minima: 18 global minima, $SH(X^*) \simeq -186.73091$.

(B_2) **(two variables)**
Definition: $B_2(x_1, x_2) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$;
Search domain: $-100 < x_j < 100, j = 1, 2$;
Local minimum: several local minima (exact number is unspecified in literature);
Global minima: $(x_1^*, x_2^*) = (0, 0), B_2(X^*) = 0$.

Easom (ES) **(two variables)**
Definition: $ES(x_1, x_2) = -\cos(x_1)\cos(x_2)\exp\left(-[(x_1 - \pi)^2 + (x_2 - \pi)^2]\right)$;

Search domain: $-100 < x_i < 100, j = 1,2$;
Local minimum: several local minima (exact number is unspecified in literature);
Global minima: $(x_1^*, x_2^*) = (\pi, \pi), ES(X^*) = -1$.

Shekel ($Shl_{4,n}$) **(four variables)**
Definition: $Shl_{4,n}(X) = -\sum_{j-1}^{m}\left(\sum_{i=4}^{4}(x_i - C_{ij})^2 + \beta_j\right)^{-1}, \beta = \frac{1}{10}[1,2,2,4,6,3,7,5,5]$,

$$C = \begin{bmatrix} 4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 5.0 & 8.0 & 6.0 & 7.0 \\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 5.0 & 1.0 & 2.0 & 3.6 \\ 4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 3.0 & 8.0 & 6.0 & 7.0 \\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 3.0 & 1.0 & 2.0 & 3.6 \end{bmatrix},$$

Search domain: $0 \leq x_j \leq 10, j = 1,2,3,4$;
Local minimum: n local minima ($n = 5,7$ or 10);
Global minima: $X^* = (4,4,4,4), Shl_{4,5}(X^*) \simeq -10.1532, Shl_{4,7}(X^*) \simeq -10.4029$, $Shl_{4,10}(X^*) \simeq -10.5364$.

Hartmann ($H_{6,4}$) **(six variables)**
Definition: $H_{6,4} = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}(x_i - p_{ij})^2\right)$
Search domain: $0 < x_j < 1, j = 1,\ldots,6$;
Local minimum: four local minimma;
Global minima: $X^* = (0.20169, 0.150011, 0.47687, 0.275332, 0.311652, 0.6573)$, $H_{6,4}(X)^* \simeq -3.322368$.

A.3 Fuzzy outputs and the errors by Shakouri and Nadimi approach [1]

The reported function in [1] is: $\widetilde{y} = (-20.08, -20.08, -20.08) + (-0.23, -0.16, -0.09)x_1 + (-1.22, -0.9, -0.58)x_2 + (1.66, 1.81, 1.96)x_3$. Applying the fuzzy arithmetic for addition and multiplication of the fuzzy numbers [167], the results are given in Table A.9.

Table A.9: The incorrect reported fuzzy output given by Shakouri and Nadimi in [1] for data set given in Table 5.7 versus the actual fuzzy outputs (taken from [6])

Obs.	Independent variable	Reported Output [1]	Actual output	Actual error
Team 1	$(2.27, 5.83, 9.39)$	$(4.98, 7.18, 9.38)$	$(4.77, 7.20, 9.63)$	2.43 (0.69)
Team 2	$(0.33, 0.85, 1.37)$	$(0.40, 0.97, 1.54)$	$(-2.72, 1.00, 4.71)$	3.20 (6.16)
Team 3	$(5.43, 13.93, 22.43)$	$(2.33, 3.94, 5.55)$	$(1.29, 3.96, 6.64)$	11.05 (1.30)
Team 4	$(1.56, 4, 6.44)$	$(1.57, 3.12, 4.67)$	$(0.56, 3.15, 5.73)$	1.56 (0.64)
Team 5	$(0.64, 1.65, 2.66)$	$(1.96, 2.87, 3.77)$	$(-0.67, 2.89, 6.46)$	2.82 (2.79)
Team 6	$(0.62, 1.58, 2.54)$	$(0.62, 1.59, 2.57)$	$(-1.58, 1.62, 4.82)$	2.24 (2.33)
Team 7	$(3.19, 8.18, 13.17)$	$(2.04, 2.65, 3.26)$	$(-1.34, 2.68, 6.70)$	7.64 (1.53)
Team 8	$(0.72, 1.85, 2.98)$	$(0.72, 1.85, 2.98)$	$(-1.09, 1.88, 4.84)$	1.83 (1.62)
T_e (RNIA (NIA))	-	(5.16)	32.78 (17.06)	32.78 (17.06)

Bibliography

[1] H. Shakouri and R. Nadimi, "A novel fuzzy linear regression model based on a non-equality possibility index and optimum uncertainty," *Applied Soft Computing*, vol. 9, no. 2, pp. 590–598, 2009.

[2] M. H. Mashinchi, M. A. Orgun, and W. Pedrycz, "Hybrid optimization with improved tabu search," *Applied Soft Computing*, vol. 11, no. 2, pp. 1993–2006, 2011.

[3] P.-T. Chang, *Fuzzy regression analysis*. PhD thesis, Kansas State University, Manhattan, Kansas, 1994.

[4] M. Hojati, C. R. Bector, and K. Smimou, "A simple method for computation of fuzzy linear regression," *European Journal of Operational Research*, vol. 166, no. 1, pp. 172–184, 2005.

[5] H. Tanaka, S. Uejima, and A. Asai, "Linear regression analysis with fuzzy model," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 12, pp. 903–907, 1982.

[6] M. H. Mashinchi, M. A. Orgun, M. Mashinchi, and W. Pedrycz, "A tabu-harmony search based approach to fuzzy linear regression," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 3, pp. 432 – 448, 2011.

[7] Y.-S. Chen, "Outliers detection and confidence interval modification in fuzzy regression," *Fuzzy Sets and Systems*, vol. 119, pp. 252–279, 2001.

[8] M. H. Mashinchi, M. A. Orgun, and M. R. Mashinchi, "A least square approach for the detection and removal of outliers for fuzzy linear regressions," in *Second World Congress on Nature & Biologically Inspired Computing*, pp. 134–139, 2010.

[9] M. H. Mashinchi, "Application of hybird optimization for spread increasing problem," tech. rep., Macquarie University, 2011.

[10] M. H. Mashinchi, L. Li, M. A. Orgun, and Y. Wang, "The prediction of trust rating based on the quality of services using fuzzy linear regression," in *IEEE International Conference on Fuzzy Systems*, pp. 1953–1959, 2011.

[11] H. Tanaka, H. Ishibuchi, and J. Watada, "Probablistic linear regression for analysis for fuzzy data," *European Journal of Operational Research*, vol. 40, pp. 389–396, 1989.

[12] B. Kim and R. R. Bishu, "Evaluation of fuzzy linear regression models by comparing membership functions," *Fuzzy Sets and Systems*, vol. 100, no. 1-3, pp. 343–352, 1998.

[13] D. H. Hong and C. Hwang, "Support vector fuzzy regression machines," *Fuzzy Sets and Systems*, vol. 138, no. 2, pp. 271–281, 2003.

[14] M. Sakawa and H. Yano, *Fuzzy linear regression and its applications*, pp. 61–80. Fuzzy Regression Analysis, Heidelberg: Physica-Verlag, 1992.

[15] E. Al-Masri and Q. H. Mahmoud, "Discovering the best web service," in *Proceedings of the 16th international conference on World Wide Web*, pp. 1257–1258, 2007.

[16] A.-R. Hedar, *Studies on Metaheuristics for Continuous Global Optimization Problems*. PhD thesis, Kyoto University, Kyoto, Japan, 2004.

[17] E. M. Montes, *Alternative Techniques to Handle Constraints in Evolutionary Optimization*. PhD thesis, Centro de Investigación y de Estudios Avanzados del Instituto Politćnico Nacional, Mexico, 2004.

[18] A. M. Connor and D. G. Tilley, "A tabu search method for the optimization of fluid power circuits," in *Proceedings of the I MECH E Part I Journal of Systems and Control Engineering*, pp. 373–381, October 1998.

[19] J. A. Bland and G. P. Dawson, "Tabu search and design optimization," *Computer Aided Design*, vol. 23, no. 3, pp. 195–201, 1991.

[20] M. Sinclair, "Comparison of the performance of modern heuristics for combinatorial optimization of real data," *Computers and Operations Research*, vol. 20, no. 7, pp. 687–695, 1993.

[21] S. Schlund, R. Muller, C. Grabmann, and B. Engels, "Conformational analysis of arginine in gas phase - a strategy for scanning the potential energy surface effectively," *Journal of Computational Chemistry*, vol. 29, no. 3, pp. 407–415, 2008.

[22] R. T. Zheng, N. Q. Ngo, P. Shum, S. C. Tjin, and L. N. Binh, "A staged continuous tabu search algorithm for the global optimization and its applications to the design of fiber bragg gratings," *Computational Optimization and Applications*, vol. 30, pp. 319–335, 2005.

[23] S.-W. Chiou, "Optimization for signal setting problems using non-smooth techniques," *Information Sciences*, vol. 179, no. 17, pp. 2985–2996, 2009.

[24] R. Chelouah and P. Siarry, "A hybrid method combining continuous tabu search and nelder-mead simplex algorithms for the global optimization of multiminima functions," *European Journal of Operational Research*, vol. 161, no. 3, pp. 636–654, 2005.

[25] S. L. Ho, S. Yang, G. Ni, and H. C. Wong, "An improved tabu search for the global optimizations of electromagnetic devices," *IEEE Transactions on Magnetics*, vol. 37, no. 5, pp. 3570–3574, 2001.

[26] A.-R. Hedar and M. Fukushima, "Tabu search by direct search methods for nonlinear global optimization," *European Journal of Operational Research*, vol. 170, no. 2, pp. 329–349, 2006.

[27] P. S. Shelokar, P. Siarry, V. K. Jayaraman, and B. D. Kulkarni, "Particle swarm and ant colony algorithms hybridized for improved continuous optimization," *Applied Mathematics and Computations*, vol. 188, no. 1, pp. 129–142, 2007.

[28] N. Mladenovic, M. Drazic, V. Kovacevic-Vujcic, and M. Cangalovic, "General variable neighborhood search for the continuous optimiation," *European Journal of Operational Research*, vol. 191, no. 3, pp. 753–770, 2008.

[29] V. V. Kovacevic-Vujcic, M. M. Cangalovic, M. D. Asic, L. Ivanovic, and M. Drazic, "Tabu search methodology in global optimization," *Computers and Mathematics with Applications*, vol. 37, no. 4, pp. 125–133, 1999.

[30] A. R. Conn, K. Scheinberg, and L. N. Vicent, *Introduction to Derivative-Free Optimization*. MPS-SIAM Series on Optimization, Philadelphia: SIAM, 2009.

[31] L. M. Rios, *Algorithms for derivative-free optimizations*. PhD thesis, University of Illinois, Urbana-Champaign, 2009.

[32] D. B. Serafini, *A framework for managing models in nonlinear optimization of computationally expensive functions*. PhD thesis, Rice University, Houston, 1999.

[33] A. L. Marsden, *Aerodynamic Noise Control by Optimal Shape Design*. PhD thesis, Stanford University, Stanford, CA, 2004.

[34] C. Audet and J. E. Dennis, Jr., "A pattern search filter method for nonlinear programming without derivatives," *SIAM journal on Optimization*, vol. 14, pp. 980–1010, April 2004.

[35] N. M. Alexandrov, J. E. Dennis, R. M. Lewis, and V. orczon, "A trust-region framework for managing the use of approximation models in optimization," *Structural and Multidisciplinary Optimization*, vol. 15, no. 1, pp. 16–23, 1998.

[36] J. C. Meza and M. L. Martinez, "Direct search methods for the molecular conformation problem," *Journal of Computational Chemistry*, vol. 15, pp. 627–632, June 1994.

[37] P. D. Frank, "Brief description of the helicopter rotor blade design project," September 1994. Informal report.

[38] S. M. Wild, *Derivative-free optimization algorithm for computationally expensive functions*. PhD thesis, Cornell University, January 2009.

[39] J. C. Meza and M. L. Martinez, "Direct search methods for the molecular conformation problem," *Journal of Computational Chemistry*, vol. 15, pp. 627–632, June 1994.

[40] P. Alberto, F. Nogueira, H. Rocha, and L. N. Vicente, "Pattern search methods for user-provided points: Application to molecular geometry problems," *SIAM Journal on Optimization*, vol. 14, no. 4, pp. 1216–1236, 2004.

[41] M. Macklem, *Low-dimensional curvature methods in derivative-free optimization on shared computing networks*. PhD thesis, Dalhousie University, Halifax, Nova Scotia, Canada, April 2009.

[42] R. Hooke and T. A. Jeeves, "Direct search solution of numerical and statistical problems," *Journal of the Association for Computing Machinery*, vol. 8, pp. 212–219, 1961.

[43] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.

[44] J. Holland, *Adaptation in Natural and Artifical Systems*. Ann Arbor: University of Michigan Press, 1975.

[45] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.

[46] F. Glover, "Tabu search: Part 1," *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190–206, 1989.

[47] M. Schonlau, *Computer experiments and global optimization*. PhD thesis, University of Waterloo, Waterloo, Ontario, Canada, 1997.

[48] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: Harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.

[49] K. Meittinen, M. M. Makela, and H. Maaranen, "Efficient hybrid methods for global continuous optimization based on simulated annealing," *Computers and Operations Research*, vol. 33, no. 4, pp. 1102–1116, 2006.

[50] C. Li, X. Liao, and J. Yu, "Tabu search for fuzzy optimization and applications," *Information Sciences*, vol. 158, pp. 3–13, 2004.

[51] R. Chelouah and P. Siarry, "Genetic and nelder-mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions," *European Journal of Operational Research*, vol. 148, no. 2, pp. 335–348, 2003.

[52] F. Glover, "Tabu search: Part 2," *ORSA Journal on Computing*, vol. 2, no. 1, pp. 4–32, 1990.

[53] N. Hu, "Tabu search method with random moves for globally optimal design," *International Journal for Numerical Methods in Engineering*, vol. 35, no. 5, pp. 1055–1070, 1992.

[54] A. M. Connor, “Memory models for improving tabu search with real continuous variables,” in *Proceedings of the 6th International Conference on Hybrid Intelligent Systems (HIS 2006)*, p. 27, November 2006.

[55] M. H. Mashinchi, M. R. Mashinchi, and M. Mashinchi, “Tabu search solution for fuzzy linear programming,” in *Proceedings of the 7th IEEE/ACIS International Conference on Computer and Information Science (IEEE/ACIS ICIS 2008)*, pp. 82–87, May 2008.

[56] K. S. Lee and Z. W. Geem, “A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice,” *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 36–38, pp. 3902–3933, 2005.

[57] M. Fesanghary, M. Mahdavi, M. M-Jolandan, and Y. Alizadeh, “Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 33–40, pp. 3080–3091, 2008.

[58] F. Romeo and A. Sangiovanni-Vincentelli, “A theoretical framework for simulated annealing,” *Algorithmica*, vol. 6, pp. 302–345, 1991.

[59] J. E. Orosz and S. H. Jacobson, “Finite-time performance analysis of static simulated annealing algorithms,” *Computational Optimization and Applications*, vol. 21, pp. 21–53, January 2002.

[60] M. D. McKay, R. J. Beckman, and C. W. J, “A comparison of three methods for selecting values of input variables in the analysis of output from a computer code,” *Technometrics*, vol. 21, pp. 239–245, 1979.

[61] P. Siarry and G. Berthiau, “Fitting of tabu search to optimize functions of continuous variables,” *International Journal for Numerical Methods in Engineering*, vol. 40, no. 13, pp. 2449–2457, 1999.

[62] R. Chelouah and P. Siarry, “Tabu search applied to global optimization,” *European Journal of Operational Research*, vol. 123, no. 2, pp. 256–270, 2000.

[63] J. Dréo and P. Siarry, “Continuous interacting ant colony algorithm based on dense heterarchy,” *Future Generation Computer Systems*, vol. 20, no. 5, pp. 841–856, 2004.

[64] K. Socha and M. Dorigo, "Ant colony optimization for continuous domains," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1155–1173, 2008.

[65] D. H. Kim, A. Abraham, and J. H. Cho, "A hybrid genetic algorithm and bacterial foraging approach for global optimization," *Information Sciences*, vol. 177, no. 18, pp. 3918–3937, 2007.

[66] I. G. Tsoulos, "Modifications of real code genetic algorithm for global optimization," *Applied Mathematics and Computation*, vol. 203, no. 2, pp. 598–607, 2008.

[67] Y.-T. Kao and E. Zahara, "A hybrid genetic algorithm and particle swarm optimization for multimodal functions," *Applied Soft Computing*, vol. 8, no. 2, pp. 849–857, 2008.

[68] E. Zahara and Y.-T. Kao, "A hybrid simplex search and particle swarm optimization for unconstrained optimization," *European Journal of Operational Research*, vol. 181, no. 2, pp. 527–548, 2007.

[69] E. Zahara and Y.-T. Kao, "Hybrid Nelder-Mead simplex search and particle swarm optimization for constrained engineering design problems," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3880–3886, 2009.

[70] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Applied Mathematics and Computation*, vol. 188, no. 2, pp. 1567–1579, 2007.

[71] E.-G. Talbi, "A taxonomy of hybrid metaheuristics," *Journal of Heuristics*, vol. 8, no. 5, pp. 541–564, 2002.

[72] M. Ji and H. Tang, "Global optimizations and tabu search based on memory," *Applied Mathematics and Computation*, vol. 159, no. 2, pp. 449–457, 2004.

[73] J. M. Machado, Y. Shiyou, S. L. Ho, and N. Peihong, "A common tabu search algorithm for the global optimization of engineering problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 190, pp. 3501–3510, 2001.

[74] M. H. Mashinchi, M. A. Orgun, and M. Mashinchi, "Solving fuzzy linear regression with hybrid optimization," in *16th International Conference Neural Information Processing*, pp. 336–343, 2009.

[75] M. M. Ali and P. Kaelo, "Improved particle swarm algorithms for global optimization," *Applied Mathematics and Computation*, vol. 196, no. 2, pp. 578–593, 2008.

[76] S.-K. S. Fan and E. Zahara, "A hybrid simplex search and particle swarm optimization for unconstrained optimization," *European Journal of Operational Research*, vol. 181, no. 2, pp. 527–548, 2007.

[77] D. Karaboga and B. Basturk, "On the performance of artificial bee colony algorithm," *Applied Soft Computing*, vol. 8, no. 1, pp. 687–697, 2008.

[78] F. O. de França, F. J. V. Zuben, and L. N. de Castro, "An artificial immune network for multimodal function optimization on dynamic environments," in *Genetic and Evolutionary Computation Conference*, pp. 289–296, 2005.

[79] J. Dréo and P. Siarry, "An ant colony algorithm aimed at dynamic continuous optimization," *Applied Mathematics and Computation*, vol. 181, no. 1, pp. 457–467, 2006.

[80] J. Dréo and P. Siarry, "Hybrid continuous interacting ant colony aimed at enhanced global optimization," *Algorithmic Operations Research*, vol. 2, no. 1, pp. 52–64, 2007.

[81] W. Tfaili and P. Siarry, "A new charged ant colony algorithm for continuous dynamic optimization," *Applied Mathematics and Computation*, vol. 197, no. 2, pp. 604–613, 2008.

[82] D. M. Jaeggi, G. T. Parks, T. Kipouros, and P. J. Clarkson, "The development of a multi-objective tabu search algorithm for continuous optimisation problems," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1192–1212, 2008.

[83] W.-L. Hung and M.-S. Yang, "An omission approach for detecting ouliers in fuzzy regression models," *Fuzzy Sets and Systems*, vol. 157, pp. 3109–3122, 2006.

[84] E. S. Lee and P.-T. Chang, "Fuzzy linear regression analysis with spread unconstrained in sign," *Computers and Mathematics with Applications*, vol. 28, no. 4, pp. 61–70, 1994.

[85] G. Peters, "Fuzzy linear regression with fuzzy intervals," *Fuzzy Sets and Systems*, vol. 63, pp. 45–55, 1994.

[86] J. Lu and R. Wang, "An enhanced fuzzy linear regression model with more flexible spreads," *Fuzzy Sets and Systems*, vol. 160, no. 17, pp. 2505–2523, 2009.

[87] S.-P. Chen and J.-F. Dang, "A variable spread fuzzy linear regression model with higher explanatory power and forecasting accuracy," *Information Sciences*, vol. 178, no. 20, pp. 3973–3988, 2008.

[88] C. Kao and C.-L. Chyu, "A fuzzy linear regression model with better explanatory power," *Fuzzy Sets and Systems*, vol. 126, no. 3, pp. 401–409, 2002.

[89] M. H. Mashinchi, M. A. Orgun, and M. Mashinchi, "Solving fuzzy linear regression with hybrid optimization," in *International Conference on Neural Information Processing*, pp. 336–343, IEEE Computer Society Press, 2009.

[90] H. Hassanpour, *A Goal Programming Approach to Fuzzy Linear Regression*. PhD thesis, Shahid Bahonar University of Kerman, Kerman, Iran, 2008.

[91] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965.

[92] L. G. O. Soto, "Interval and fuzzy regressions to enhance drastic aquifer vulnerability assessments," Master's thesis, Texas A&M University, Kingsville, December 2007.

[93] L. T. Tran, *Multi-objective fuzzy regression and its application to erosion prediction*. PhD thesis, University of Hawaii, August 1999.

[94] E. C. Ozelkan, *Multi-objective fuzzy regression applied to the calibration of conceptual rainfall-runoff models*. PhD thesis, University of Arizona, 1997.

[95] D. Dubois and H. Prade, *Fuzzy Sets & Systems: Theory and Applications*. Academic Press, 1980.

[96] A. Kaufman and M. M. Gupta, *Introduction to fuzzy arithmetic, theory and application*. New York: Van Nostrand Reinhold Company, 1991.

[97] P. Diamond, "Fuzzy least squares," *Information Sciences*, vol. 46, no. 3, pp. 141–157, 1988.

[98] A. Bardossy, I. Bogardi, and L. Duckstein, "Fuzzy regression in hydrology," *Water Resources Research*, vol. 26, no. 7, pp. 1497–1508, 1990.

[99] L.-H. Chen and C.-C. Hsueh, "A mathematical programming method for formulating a fuzzy regression model based on distance criterion," *IEEE Transactions on Systems, Man, and Cybernetics-part B: Cybernetics*, vol. 37, no. 3, pp. 705–712, 2007.

[100] P. D. 'Urso, "Linear regression analysis for fuzzy/crisp input and fuzzy/crisp output data," *Computational Statistics & Data Analysis*, vol. 42, no. 1-2, pp. 47–72, 2003.

[101] C. Kao and C.-L. Chyu, "Least-squares estimates in fuzzy regression analysis," *European Journal of Operational Research*, vol. 148, no. 2, pp. 426–435, 2003.

[102] A. Bisserier, R. Boukezzoula, and S. Galichet, "A revisited approach to linear fuzzy regression using trapezoidal fuzzy intervals," *Information Sciences*, vol. 180, 2010.

[103] A. R. Arabpour and M. Tata, "Estimating the parameters of a fuzzy linear regression model," *Iranian Journal of Fuzzy Systems*, vol. 5, no. 2, pp. 1–19, 2008.

[104] P. D. 'Urso and A. Santoroa, "Fuzzy clusterwise linear regression analysis with symmetrical fuzzy output variable," *Computational Statistics & Data Analysis*, vol. 51, no. 1, pp. 287–313, 2006.

[105] T. Feuring, W. Golubski, and M. Gassmann, "Fuzzy regression: a genetic programming approach," in *Proceedings of Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies*, (Brighton, UK), pp. 349–352, 2000.

[106] R. M. Dom, *Fuzzy regression model for the prediction of oral cancer susceptibility*. PhD thesis, University of Malaya, Kuala Lumpure, Malaysia, 2009.

[107] Y.-H. O. Chang and B. M. Ayyub, "Fuzzy regression methods - a comparative assessment," *Fuzzy Sets and Systems*, vol. 119, no. 2, pp. 187–203, 2001.

[108] M.-S. Yang and C.-H. Ko, "On cluster-wise fuzzy regression analysis," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 27, no. 1, pp. 1–13, 1997.

[109] A. Bargiela, W. Pedrycz, and T. Nakashima, "Multiple regression with fuzzy data," *Fuzzy Sets and Systems*, vol. 158, no. 19, pp. 2169–2188, 2007.

[110] R. A. Aliev, B. Fazlollahi, and R. M. Vahidov, "Genetic algorithms-based fuzzy regression analysis," *Soft Computing*, vol. 6, no. 6, pp. 470–475, 2002.

[111] D. T. Redden and W. H. Woodall, "Further examination of fuzzy linear regression," *Fuzzy Sets and Systems*, vol. 79, pp. 203–211, 1996.

[112] W. Näther, "Regression with fuzzy random data," *Computational Statistics & Data Analysis*, vol. 51, no. 1, pp. 235–252, 2006.

[113] P.-Y. Hao and J.-H. Chiang, "Fuzzy regression analysis by support vector learning approach," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 2, pp. 428–441, 2008.

[114] D. H. Hong and C. Hwang, "Interval regression analysis using quadratic loss support vector machine," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 2, pp. 229–237, 2005.

[115] R. Coppi, "Management of uncertainty in statistical reasoning: The case of regression analysis," *International Journal of Approximate Reasoning*, vol. 47, no. 3, pp. 284–305, 2008.

[116] J. J. Buckley, "Fuzzy prediction in linear regression," *Studies in Fuzziness and Soft Computing*, vol. 196, pp. 177–179, 2006.

[117] S. Imoto, Y. Yabuuchi, and J. Watada, "Fuzzy regression model of R&D project evaluation," *Applied Soft Computing*, vol. 8, no. 3, pp. 1266–1273, 2008.

[118] J. A. Sánchez, "Calculating insurance claim reserves with fuzzy regression," *Fuzzy Sets and Systems*, vol. 157, no. 23, pp. 3091–3108, 2006.

[119] P. Diamond, *Least squares fitting of several fuzzy variables*, pp. 329–331. Analysis of Fuzzy Information, Tokyo: CRC Press, August 1987.

[120] S. Josef, "on the effect of linear data transformations in possibilistic fuzzy linear regression," *fuzzy sets and systems*, vol. 45, pp. 185–188, 1992.

[121] C.-T. Yeh, "Reduction to least-squares estimates in multiple fuzzy regression analysis," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 4, pp. 935–948, 2009.

[122] C.-T. Yeh, "A formula for fuzzy linear regression analysis," in *IEEE international conference on fuzzy systems*, pp. 2845–2850, 2011.

[123] L.-H. Chen and C.-C. Hsueh, "Fuzzy regression models using the least-squares method based on the concept of distance," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 6, pp. 1259–1272, 2009.

[124] J. Chachi and T. S. Mahmoud, "An interval-based approach to fuzzy regression for fuzzy input-output data," in *IEEE international conference on fuzzy systems*, pp. 2859–2863, 2011.

[125] S. M. Taheri, "Trends in fuzzy statistics," *Austrian Journal of Statistics*, vol. 32, no. 3, pp. 239–257, 2003.

[126] A. Celmins, "Least squares model fitting to fuzzy vector data," *Fuzzy Sets and Systems*, vol. 22, no. 3, pp. 245–269, 1987.

[127] K. S. Lee and Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 36-38, pp. 3902–3933, 2005.

[128] M. G. H. Omran and M. Mahdavi, "Global-best harmony search," *Applied Mathematics and Computation*, vol. 198, no. 2, pp. 643–656, 2008.

[129] M. H. Mashinchi, M. R. Mashinchi, and M. Mashinchi, "A two-phased genetic algorithm approach for linear and quadratic fuzzy regression," *Set-valued Mathematics and Applications*, vol. 1, no. 2, pp. 183–190, 2008.

[130] K. Y. Chan, C. K. Kwong, and Y. C. Tsim, "A genetic programming based fuzzy regression approach to modelling manufacturing processes," *International Journal of Production Research*, vol. 48, no. 7, pp. 1967–1982, 2009.

[131] N. Kasabov, "Evolving intelligence in humans and machines: Integrative evolving connectionist systems approach," *IEEE Computational Intelligence Magazine*, vol. 3, no. 3, pp. 23–37, 2008.

[132] M. R. Mashinchi and A. Selamat, "An improvement on genetic-based learning method for fuzzy artificial neural networks," *Applied Soft Computing*, vol. 9, no. 4, pp. 1208–1216, 2009.

[133] M. J. Watts, "A decade of Kasabovs evolving connectionist systems: A review," *IEEE Transations on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, vol. 39, no. 3, pp. 253–269, 2009.

[134] A. Abdalla and J. J. Buckley, "Monte Carlo methods in fuzzy linear regression II," *Soft Computing*, vol. 12, no. 5, pp. 463–468, 2008.

[135] M. Modarres, E. Nasrabadi, and M. M. Nasrabadi, "Fuzzy linear regression models with least square errors," *Applied Mathematics and Computation*, vol. 163, no. 2, pp. 977–989, 2005.

[136] M. M. Nasrabadi and E. Nasrabadi, "A mathematical-programming approach to fuzzy linear regression analysis," *Applied Mathematics and Computation*, vol. 55, no. 3, pp. 873–881, 2004.

[137] D. Savic and W. Pedrycz, "Evolution of fuzzy linear regression models," *Fuzzy Sets and Systems*, vol. 39, pp. 51–63, 1991.

[138] E. C. Özelkan and L. Duckstein, "Multi-objective fuzzy regression: a general framework," *Computers & Operations Research*, vol. 27, no. 7-8, pp. 635–652, 2000.

[139] C. Kao and P.-H. Lin, "Entropy for fuzzy regression analysis," *International Journal of Systems Science*, vol. 36, no. 14, pp. 869–876, 2005.

[140] C.-C. Yao and P.-T. Yu, "Fuzzy regression based on asymmetric support vector machines," *Applied Mathematics and Computation*, vol. 182, no. 1, pp. 175–193, 2006.

[141] M.-S. Yang and T.-S. Lin, "Fuzzy least-squares linear regression analysis for fuzzy input-output data," *Fuzzy Sets and Systems*, vol. 126, no. 3, pp. 389–399, 2002.

[142] M. Modarres, E. Nasrabadi, and M. M. Nasrabadi, "Fuzzy linear regression analysis from the point of view risk," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 12, no. 5, pp. 635–649, 2004.

[143] B. Gladysz and D. Kuchta, "Outliers detection in selected fuzzy regression models," in *Proceedings of the 7th international workshop on Fuzzy Logic and Applications*, pp. 211–218, 2007.

[144] K. Y. Chan, C. K. Kwong, and T. C. Fogarty, “Modelling manufacturing processes using a genetic programming-based fuzzy regression with detection of outliers,” *Information Sciences*, vol. 180, pp. 506–518, 2010.

[145] P. D. ’Urso, “Linear regression analysis for fuzzy/crisp input and fuzzy/crisp output data,” *Computational Statistics and Data Analysis*, vol. 42, no. 1-2, pp. 47–72, 2003.

[146] L.-H. Chen and C.-C. Hsueh, “A mathematical programming method for formulating a fuzzy regression model based on distance criterion,” *IEEE Transations on System, Man and Cybernetics, Part B*, vol. 37, no. 3, pp. 705–712, 2007.

[147] J. J. Buckley and T. Feuring, “Linear and non-linear fuzzy regression: evolutionary algorithm solutions,” *Fuzzy Sets and Systems*, vol. 112, no. 3, pp. 381–394, 2000.

[148] L.-H. Chen and C.-C. Hsueh, “Fuzzy regression models using the least squares method based on the concept of distance,” *IEEE Transations on Fuzzy Systems*, vol. 17, no. 6, pp. 1259–1272, 2009.

[149] L. Coelho, “A quantum particle swarm optimizer with chaotic mutation operator,” *Chaos, Solitons & Fractals*, vol. 37, pp. 1409–1418, 2008.

[150] C. A. C. Coello, “Use of a self-adaptive penalty approach for engineering optimization problems,” *Computers in Industry*, vol. 41, no. 2, pp. 113–127, 2000.

[151] Q. He and L. wang, “An effective co-evolutionary particle swarm optimization for constrained engineering design problems,” *Engineering Applications of Artificial Intelligence*, vol. 20, no. 1, pp. 89–99, 2007.

[152] B. K. Kanna and S. N. Kramer, “An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design,” *Transactions of the ASME, Journal of Mechanical Design*, vol. 116, pp. 318–320, 1994.

[153] M. Mahdavi, M. Fesanghary, and E. Damangir, “An improved harmony search algorithm for solving optimization problems,” *Applied Mathematics and Computation*, vol. 188, no. 2, pp. 1567–1579, 2007.

[154] L. Li, Y. Wang, and E.-P. Lim, “Trust-oriented composite service selection and discovery,” in *International Conference on Service Oriented Computing*, pp. 50–67, 2009.

[155] L. Li, Y. Wang, and E.-P. Lim, “Trust-oriented composite service selection with QoS constraints,” *Journal of Universal Computer Science*, vol. 16, no. 13, pp. 1720–1744, 2010.

[156] L. Li and Y. Wang, “Subjective trust inference in composite services,” in *AAAI Conference on Artificial Intelligence*, pp. 1377–1384, 2010.

[157] A. Jøsang, R. Ismail, and C. Boyd, “A survey of trust and reputation systems for online service provision,” *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, 2007.

[158] Y. Wang and E.-P. Lim, “The evaluation of situational transaction trust in e-service environments,” in *International Conference of Engineering and Business Education*, pp. 265–272, 2008.

[159] G. Zacharia and P. Maes, “Trust management through reputation mechanisms,” *Applied Artificial Intelligence*, vol. 14, no. 9, pp. 881–907, 2000.

[160] L. Xiong and L. Liu, “PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, 2004.

[161] L.-H. Vu, M. Hauswirth, and K. Aberer, “QoS-based service selection and ranking with trust and reputation management,” in *nternational Conference on Cooperative Information Systems*, pp. 466–483, 2005.

[162] L. Li, Y. Wang, and V. Varadharajan, “Fuzzy regression based trust prediction in service-oriented applications,” in *Proceedings of the 6th International Conference on Autonomic and Trusted Computing*, pp. 221–235, 2009.

[163] R. Mohanty, V. Ravi, and M. R. Patra, “Web-services classification using intelligent techniques,” *Expert Systems with Applications*, vol. 37, no. 7, pp. 5484–5490, 2010.

[164] E. Al-Masri and Q. H. Mahmoud, “Investigating web services on the world wide web,” in *Proceeding of the 17th international conference on World Wide Web*, pp. 795–804, 2008.

[165] M. Alrifai, T. Risse, P. Dolog, and W. Nejdl, "A scalable approach for qos-based web service selection," in *Proceedings of the International Conference on Service Oriented Computing*, pp. 190–199, 2008.

[166] L. Wei and M. Zhao, "A niche hybrid genetic algorithm for global optimization of continuous multimodal functions," *Applied Mathematics and Computation*, vol. 160, no. 3, pp. 649–661, 2005.

[167] G. J. Klir and B. Yuan, *Fuzzy sets and fuzzy logic: theory and applications*. Prentic Hall, 1995.