Variable Dependence Interaction and Multi-objective Optimisation

Ashutosh Tiwari and Rajkumar Roy

Department of Enterprise Integration,
School of Industrial and Manufacturing Science,
Cranfield University, Cranfield, Bedford,
MK43 OAL, United Kingdom (UK).
E-mail: {a.tiwari, r.roy}@cranfield.ac.uk
Tel: +44 (0) 1234 754072, Fax: +44 (0) 1234 750852

Abstract

Interaction among decision variables is inherent to a number of real-life engineering design optimisation problems. There are two types of interaction that can exist among decision variables: inseparable function interaction and variable dependence. The aim of this paper is to propose an Evolutionary Computing (EC) technique for handling variable dependence in multi-objective optimisation problems. In spite of its immense potential for real-life problems, lack of systematic research has plagued this field for a long time. The paper attempts to fill this gap by devising a definition of variable dependence. It then uses this analysis as a background for identifying the challenges that variable dependence poses for optimisation algorithms. The paper further presents a brief review of techniques for handling variable dependence in optimisation problems. Based on this analysis, it devises a solution strategy and proposes an algorithm that is capable of handling variable dependence in multi-objective optimisation problems. The working of the proposed algorithm is demonstrated, and its performance is compared to that of two high performing evolutionary-based multi-objective optimisation algorithms, NSGA-II and GRGA, using two test problems extracted from literature. The paper concludes by giving the current limitations of the proposed algorithm and the future research directions.

1 INTRODUCTION

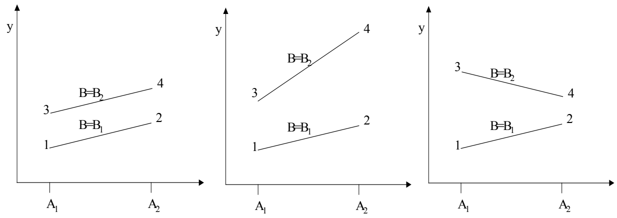
Real-life engineering design optimisation problems, as opposed to the theoretical problems (test cases), are those that are encountered in industry. Some examples of these problems are the design of aerospace structures for minimum weight, the surface design of automobiles for improved aesthetics and the design of civil engineering structures for minimum cost (Rao, 1996). A survey of industry and literature reveals that along with multiple objectives, constraints, qualitative issues and lack of prior knowledge, most real-life design optimisation problems also involve interaction among decision variables (Roy et al., 2000). However, lack of systematic research has plagued the field of interaction for a long time. This can mainly be attributed to the lack of sophisticated techniques, and inadequate hardware and software technologies. However, in the last two decades, with the improvements in hardware and software technologies some research has been carried out in this area especially in the field of statistical data analysis (Draper and Smith, 1998). This has been further augmented in the recent past with the growth of computational intelligence techniques like Evolutionary Computing (EC), Neural Networks (NNs) and Fuzzy Logic (FL) (Pedrycz, 1998). This paper focuses on the development of an evolutionary-based algorithm for handling variable interaction in multi-objective optimisation problems.

2 TYPES OF VARIABLE INTERACTION

In an ideal situation, desired results could be obtained by varying the decision variables of a given problem in a random fashion independent of each other. However, due to interaction this is not possible in a number of cases, implying that if the value of a given variable changes, the values of others should be changed in a unique way to get the required results. The two types of interaction that can exist among decision variables are discussed below.

2.1 INSEPARABLE FUNCTION INTERACTION

The first type of interaction among decision variables, known as inseparable function interaction, is discussed in detail by Tiwari et al. (2001). This interaction occurs when the effect that a variable has on the objective function depends on the values of other variables in the function (Taguchi, 1987). This concept of interaction can be understood from Figure 1.



(a) (b) (c)

Figure 1: Examples of Interaction (a) No Interaction (b) Synergistic Interaction (c) Anti-synergistic Interaction (Phadke, 1989)

In GA literature, the inseparable function interaction, as defined above, is termed as epistasis. The GA community defines epistasis as the interaction between different genes in a chromosome (Beasley et al., 1993). A review of literature reveals that the evolutionary-based techniques for handling inseparable function interaction can be classified into two broad categories based on the approach used for the prevention of building block disruption. These categories involve managing the race between linkage evolution and allele selction (Harik, 1997), and modelling the promising solutions (Muhlenbein and Mahnig, 1999).

A number of real-life examples can be found in literature that involve this type of interaction. For example, the temperature (T) of an ideal gas varies with its pressure (P) and volume (V) as T=kPV, where k is the constant of proportionality. This equation has cross-product term PV clearly demonstrating the interaction between P and V in the definition of T.

2.2 VARIABLE DEPENDENCE

The second type of interaction among decision variables, known as variable dependence, is the main focus of this paper. This interaction occurs when the variables are functions of each other, and hence cannot be varied independently. Here, change in one variable has an impact on the value of the other. A typical example of this type of interaction is the case when the function *y* is A^2+B^2, where *A* and *B* are as defined below.

$$A = Random(a, b)$$
$$B = f(A) + Random(c, d)$$

As can be seen, variable *A* is fully independent and can take any random value between *a* and *b*. On the other hand, variable *B* is not fully independent and has two components. The first component, which is a function of variable *A*, takes values depending on the values of *A*. The second component is a random number lying between *c* and *d*. It should be noted that in case of no dependence among decision variables, *a* and *b* define the range of variable *A*, and *c* and *d* define the range of variable *B*.

The above example reveals that the presence of dependence among decision variables has the following effects.

- Both variables *A* and *B* cannot simultaneously take random values in their respective ranges. If variable *A* takes a value A_1, variable *B* can take only those random values that lie between $[f(A_1)+c]$ and $[f(A_1)+d]$. With the change in value of *A*, the range of random values that *B* can take also changes. So, the variables cannot be varied independently of each other.
- The above discussion implies that the presence of dependence among decision variables modifies the shape and location of variable search space. In case of no dependence among decision variables, both variables *A* and *B* can independently take random values in their respective ranges making the *A-B* search space rectangular in shape. However, the presence of dependence makes the search space take the shape and location based on the nature of function *f(A)*.

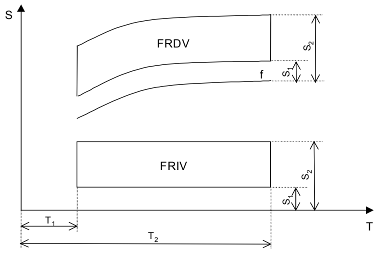


Figure 2: Relationship between Stress(S) and Temperature(T)
(FRIV: Feasible Region with Independent Variables and FRDV: Feasible Region with Dependent Variables)

The dependence among decision variables is frequently observed in real-life problems. As an example, the resistance (*R*) of a wire is defined in terms of two variables, namely Temperature (*T*) and Stress (*S*), where *T* and *S* are as defined below.

$$R = F(S, T)$$
$$T = Random(T_1, T_2)$$
$$S = f(T) + Random(S_1, S_2)$$

This real-life problem is analogous to the example discussed earlier. As illustrated in Figure 2, the presence of dependence among decision variables modifies the variable search space. In case of no dependence among decision variables, *T-S* search space is rectangular in shape. It is shown as FRIV (Feasible Region with Independent Variables) in Figure 2. In presence of dependence among variables, the modified search space is shown as FRDV (Feasible Region with Dependent Variables) in Figure 2.

3 CHALLENGES POSED BY VARIABLE DEPENDENCE

Complex variable dependence poses a number of challenges for multi-objective optimisation algorithms. In the presence of variable dependence, the decision variables cannot be varied independently of each other. Also, the search space gets modified creating a new feasible region based on the dependence among decision variables. This is demonstrated in Figure 2. Depending upon the nature of variable dependency, additional features (such as bias (non-linearity), multi-modality, deception and discontinuity) may also be introduced in the problem. A generic Genetic Algorithm (GA) independently varies the decision variables and works in the feasible region that does not take variable dependence into account. So, it creates solutions that have limited practical significance since they do not lie in the actual feasible region of the search space. Therefore, there is a need to develop GAs that have mechanisms for handling variable dependence in their search processes.

4 TECHNIQUES FOR HANDLING VARIABLE DEPENDENCE

Most of the dependent-variable optimisation problems do not have known dependency relationships. In these problems, multiple sets of variable values are available from which the dependency relationships need to be inferred. An optimisation algorithm that is capable of handling variable dependence should be able to infer these relationships from the given data, identify the independent variables and manage the search process accordingly. Due to the lack of systematic research in this area, the literature in the field of optimisation does not report any dedicated technique that can deal with variable dependence. However, as shown in Table 1, the survey of literature in related areas of research reveals some techniques that can be used for inferring dependency relationships among decision variables and identifying independent variables.

Table 1: Techniques for Identification of Dependency Relationships and Independent Variables

Identification of Dependency Relationships	• Neural Networks (NNs) (Hertz et al., 1991; Richards, 1998; Gershenfeld, 1999) • Probabilistic Modelling (PM) (Pelikan et al., 1998; Larranaga et al., 1999; Evans and Olson, 2000; Muhlenbein and Mahnig, 1999) • Regression Analysis (RA) (Frees, 1996; Draper and Smith, 1998; Evans and Olson, 2000)
Identification of Independent Variables	• Tree Diagrams (TDs) (Banzhaf et al., 1998; Richards, 1998; Larranaga et al., 1999) • Direct Analysis (DA) (Gershenfeld, 1999)

4.1 IDENTIFICATION OF DEPENDENCY RELATIONSHIPS

Table 2 presents an analysis of the techniques that can be used for inferring dependency relationships from the avaiable sets of variable values. This table highlights the following.

- NNs: As can be seen from Table 2, the NNs require a priori knowledge regarding the classification of variables as dependent and independent (Hertz et al., 1991). Since this information is rarely available in real-life problems, the choice of the NNs is ruled out in spite of their other attractive features.

Table 2: Analysis of Techniques for Identification of Dependency Relationships

Comparative Analysis		Techniques for Identification of Dependency Relationships		
		Regression Analysis (RA)	**Neural Networks (NNs)**	**Probabilistic Modelling (PM)**
Features	**Difficulty of Implementation**	Medium	High	Very high (due to many open issues)
	Accuracy	Dependent on degree of RA equation	Dependent on number of hidden units	Dependent on choice of modelling method
	Computational Expense	Low	High	Medium
	Nature of Dependency Relationships	Explicit	Explicit (for given dependent variables)	Purely implicit
	Identification of Multiple Dependency Relationships	Multiple RA equations	Built-in multiple relationships (based on choice of NN structure)	Built-in multiple relationships
	Identification of Independent Variables	Through multiple repetitions of RA	Not possible	Not required
	Difficulty of Data Addition	Medium (repetition required)	Medium (repetition required by most NNs)	Low (updating required)

- PM: PM is also a very powerful technique, requiring little information regarding the nature of variables. As shown in Table 2, it also has a number of other features that are required for dealing with real-life problems. However, the application of PM to model multiple interacting decision variables is a relatively new area of research, and a number of research issues need to be addressed before it could be chosen for handling real-life problems having multiple real variables (Evans and Olson, 2000).
- RA: Table 2 reveals that the multiple explicit equations that are identified by the RA give good insight to the designer regarding the relationships among decision variables. RA is also easy to implement and maintain (Frees, 1996). Further, it addresses most of the above-mentioned limitations of NNs and PM. However, the accuracy of RA is dependent on its degree.

4.2 IDENTIFICATION OF INDEPENDENT VARIABLES

The main strengths and weaknesses of the techniques used for the identification of independent variables are the following.

- TDs: The dependence among decision variables can be graphically represented using TDs, in which each node represents a variable in the problem. TDs are easy to use and have good visualisation capabilities, but they are difficult to be encoded in a computer language.
- DA: DA involves the analysis of dependency equations to identify the independent variables. This method is easy to be encoded in a computer language but is difficult to visualise.

5 PROPOSED GA FOR VARIABLE DEPENDENCE (GAVD)

This section proposes a novel algorithm 'GA for Variable Dependence (GAVD)', described in Figure 3. Based on the discussion in Section 4, the RA is chosen in GAVD to identify variable dependency equations using the data provided. Furthermore, GAVD uses TDs for visualisation of dependency relationships, and DA to automate the identification of independent variables. The steps involved in GAVD are described below.

5.1 STEP 1: IDENTIFICATION OF DEPENDENCY RELATIONSHIPS

This step is omitted in those cases in which the dependency relationships are known. In the other cases, this step analyses the given data for identifying multiple dependency equations, while keeping the computational expense as low as possible. GAVD uses RA in such a way that it not only identifies all non-decomposable relationships among decision variables but also removes any cyclic dependency in those relationships. To attain this, a strategy that ensures good 'book keeping' is adopted. The salient features of this strategy are discussed below.

- The RA that is used in GAVD breaks down a regression equation until it becomes non-decomposable. In this way, all the underlying relationships among the decision variables are identified.
- A Dependency Chart (DC), which is a tool for DA, is maintained to keep track of the variables that are identified as dependent (D) and independent (I) in the regression process. In this way, unnecessary repetitions of RA are avoided for the variables that have already been identified as 'D' or 'I'. This also ensures that the regression equations do not involve any cyclic dependency.
- When determining the regression equation for a given variable, only those variables that are marked as 'I' or are unmarked in DC are considered as independent. This guarantees that the variables that are identified as 'D' are not considered as independent in subsequent stages of the RA, thereby ensuring that the regression equations obtained are as non-decomposable as possible. This also reduces the number of variables that are considered at each stage of the RA.

5.2 STEP 2: IDENTIFICATION OF INDEPENDENT VARIABLES

TDs are used here for visual representation of relationships among decision variables. A TD is constructed here to give a visual representation of the dependency relationships to the user. The end nodes of this tree are the independent variables. The TD also aids in the identification of cyclic dependencies that may be present in the given dependency equations. Since TDs are difficult to be encoded in a computer language, the DC is used to automate the process of identification of independent variables and remove any cyclic dependency. Here, the DC is used to identify the independent variables as those that are marked as 'I'. The construction of this chart also aids the identification and removal of cyclic dependencies from the dependency equations.

5.3 STEP 3: OPTIMISATION

Being a high-performing latest algorithm, Generalised Regression GA (GRGA) has been chosen as the optimisation engine for GAVD. GRGA is a multi-objective optimisation algorithm that uses RA for handling complex inseparable function interaction (Tiwari et al., 2001). Here, the independent variables, identified in the previous step, define the GA chromosome. For each alternative solution generated by the GA, the dependency equations are used to calculate the values of the dependent variables. It should be noted here that the bounds on independent variables are treated as variable limits and those on dependent variables are treated as constraints.

Since GAVD uses GRGA as its optimisation engine, the basic operations of GRGA also form part of GAVD. In addition, it uses the RA to model the relationship among decision variables. Therefore, the overall computational complexity of GAVD is the complexity of GRGA increased with the complexity of the RA, where in most cases the latter is much smaller than the former.

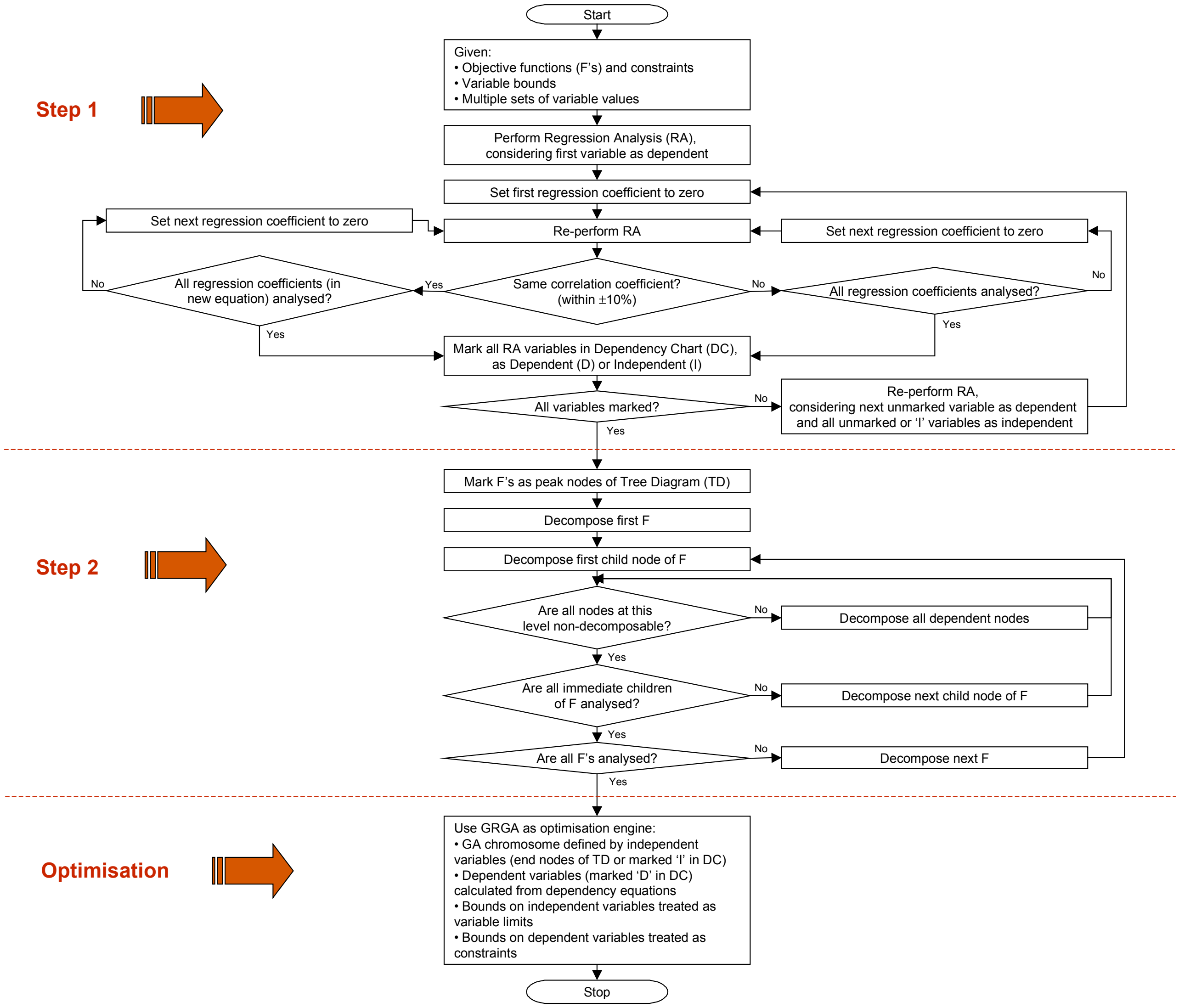


Figure 3: GA for Variable Dependence (GAVD)

5.4 A WORKED EXAMPLE

This worked example demonstrates the application of GAVD to a problem that has dependence among its decision variables. This problem is given below.

$$Objective\_Function : F = F(x_1, x_2, x_3, x_4, x_5)$$
$$\forall x_i^{(L)} \le x_i \le x_i^{(U)}, i = 1 \ldots 5$$
$$Given : Multiple\_Sets\_of\_Variable\_Values$$

Suppose the underlying relationships among decision variables that need to be identified are as follows.

$$x_1 = f_1(x_2, x_3)$$
$$x_3 = f_2(x_2, x_4, x_5)$$

The flowchart of Figure 3 identifies the following steps for solving this problem.

- Determine the following equation for x_1.

  $$x_1 = v_1(x_2, x_3, x_4, x_5)$$

- No change is observed in correlation coefficient, when the RA is performed with the regression coefficient of x_2 set to zero. The new equation is as follows.

$x_1 = v_1'(x_3, x_4, x_5)$

- Correlation coefficients reduce, when the RA is performed with the regression coefficients of x_3, x_4 and x_5 set to zero in steps.
- Mark x_1 as 'D' and x_3, x_4 and x_5 as 'I' in the DC (Table 3).
- Determine the following equation for x_2 in terms of those variables that are so far identified as 'I' or are so far unmarked in the DC.

$x_2 = v_2(x_3, x_4, x_5)$

- Correlation coefficients reduce, when the RA is performed with the regression coefficients of x_3, x_4 and x_5 set to zero in steps.
- Mark x_2 as 'D' and x_3, x_4 and x_5 as 'I' in the DC (Table 3).
- The variables marked 'I' in the DC are independent whereas those marked 'D' are dependent.
- Use the dependency equations determined above for drawing the TD for the problem (Figure 4). The nodes that are encircled in this figure represent the independent variables. All other variables are treated as dependent.
- Use GRGA as the optimisation engine.

➢ *x_3, x_4 and x_5 constitute the GA chromosome.*

➢ *x_1 and x_2 are determined from the dependency equations.*

➢ *Bounds on x_3, x_4 and x_5 are treated as variable limits.*

➢ *Bounds on x_1 and x_2 are treated as constraints.*

Table 3: Dependency Chart (DC) for Worked Example

Dependency Chart (DC)		**Variables**				
		X₁	***X₂***	***X₃***	***X₄***	***X₅***
Regression Equations	***X₁***	D		I	I	I
	X₂		D	I	I	I
	X₃					
	X₄					
	X₅					

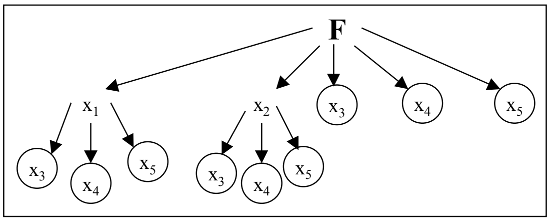


Figure 4: Tree Diagram (TD) for Worked Example

Table 4: Test Problems for Performance Analysis of GAVD

Problem	**Objective Functions (Minimisation)**	**Dependency Equations**
Problem-1	$D(\vec{x}') = \frac{1}{(1-\exp(-4))}\left[1-\exp(-4x_1)\right], \forall 0 \le x_1 \le 1$ $I(\vec{x}'') = 2 - \exp(-2x_2)\cos(8\pi x_2), \forall 0 \le x_2 \le 1$ $s(f_1, I) = 2 - (f_1 / I)^{0.6}$ $f_1 = D(\vec{x}')$ $f_2 = s(f_1, I) \times I(\vec{x}'')$	$x_2 = 1 - 0.1x_3 - 0.2x_3^2 - 0.3x_4 - 0.1x_4^2 - 0.3x_3x_4$ $\forall 0 \le x_3 \le 1, \forall 0 \le x_4 \le 1$ $Data\_Generation : x_2' = x_2 + Normal(0, 0.05)$ (Figure 5(a))
Problem-2	$D(\vec{x}') = \frac{1}{(1-\exp(-3))}\left[1-\exp(-3x_1)\right], \forall 0 \le x_1 \le 1$ $I(\vec{x}'') = 3 - \exp(-x_2)\cos(2\pi x_2) - \exp(-x_3)\cos(4\pi x_3), \forall 0 \le x_2, x_3 \le 1$ $s(f_1, I) = 2 - (f_1 / I)^{0.4} - (f_1 / I)\cos(8\pi f_1^2)$ $f_1 = D(\vec{x}')$ $f_2 = s(f_1, I) \times I(\vec{x}'')$	$x_2 = 0.2 + 0.2x_3 + 0.6x_3^2, \forall 0 \le x_3 \le 1$ $Data\_Generation : x_2' = x_2 + Normal(0, 0.05)$ (Figure 5(b))

6 PERFORMANCE ANALYSIS

In this section, GAVD is tested using two multi-objective optimisation test problems that have dependence among their decision variables (Table 4). The features of these test problems make them particularly difficult for multi-objective optimisation algorithms. In the absence of any dedicated technique for handling variable dependence, this section compares the performance of GAVD with two high-performing multi-objective optimisation algorithms: NSGA-II and GRGA. However, unlike GAVD, both these algorithms do not take variable dependency into account.

6.1 EXPERIMENTAL RESULTS

All the tests reported here correspond to 100 population size, 500 generations, 0.8 crossover probability, 0.05 mutation probability, and simulated binary crossover with 10 crossover distribution index and 50 mutation distribution index. The results obtained from these tests are shown in Figure 6 for Problem-1 and Figure 7 for Problem-2. The γ (convergence metric) and Δ (diversity metric) values corresponding to these results are shown in Table 5 (Deb et al., 2000). These results form the typical set obtained from 10 runs with different random number seed values. No major variation was observed in the results with the change in the seed values.

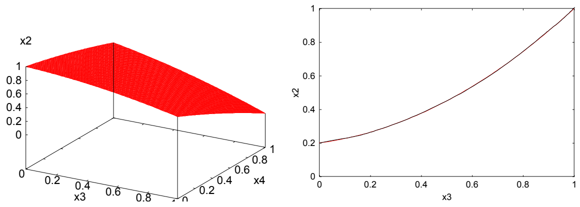


Figure 5: Dependency Relationships (a) Problem-1 (b) Problem-2

Table 5: Performance Metrics in Problems 1 and 2

Performance Metrics		Problem-1		Problem-2	
		γ	Δ	γ	Δ
Optimisation Algorithms	NSGA-II	1.209567	0.090002	0.986345	0.083956
	GRGA	0.009143	0.080121	1.654703	0.045431
	GAVD	0.008221	0.081124	0.001373	0.014564

6.2 DISCUSSION OF RESULTS

The following observations can be made from the results obtained from Problem-1 (Figure 6, Table 5).

- Since the dependency equation covers the full range of x_2, it does not alter the Pareto front. Therefore, the Pareto fronts for the original problem (with no dependence) and the dependent-variable problem coincide with each other.
- GRGA and NSGA-II do not incorporate variable dependence in their solution strategies. However, since the original and the new Pareto fronts are coincident in this case, the GRGA is able to locate the Pareto front. However, NSGA-II gets trapped in one of the local fronts.
- The dependency equation is quadratic, making it possible for the GAVD (that uses quadratic RA) to exactly model the dependence. Hence, the Pareto front that the GAVD sees coincides with the true Pareto front. Furthermore, since GAVD uses GRGA as its optimisation engine, it is able to converge to the Pareto front and distribute the solutions uniformly across the front.

The following observations can be made from the results obtained from Problem-2 (Figure 7, Table 5).

- In this problem, the original Pareto front occurs when both x_2 and x_3 are equal to 0. Due to the given dependency among these variables, this is no longer possible. This causes modifications in the search space and the Pareto front.
- GRGA converges to the global Pareto front of the original problem (with no dependence among its decision variables). However, since the new Pareto front does not coincide with the original one, the results from GRGA are not feasible in this case. Similar to the previous case, NSGA-II gets trapped on a local front, which in this case coincidentally lies in the new search space.
- Also, since GAVD uses quadratic RA, it is able to exactly determine the dependency equation in this case. Hence, the Pareto front seen by GAVD is the same as that of the actual dependent-variable problem. Therefore, GAVD converges to the Pareto front and distributes the solutions uniformly across the front.

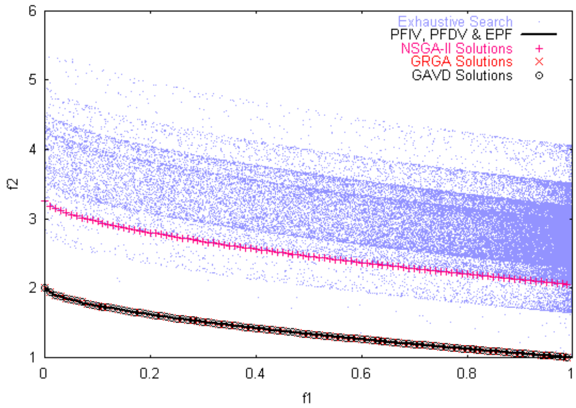


Figure 6: GAVD Performance in Problem-1 (PFIV: Pareto Front for Independent Variables, PFDV: Pareto Front for Dependent Variables, EPF: Estimated Pareto Front)

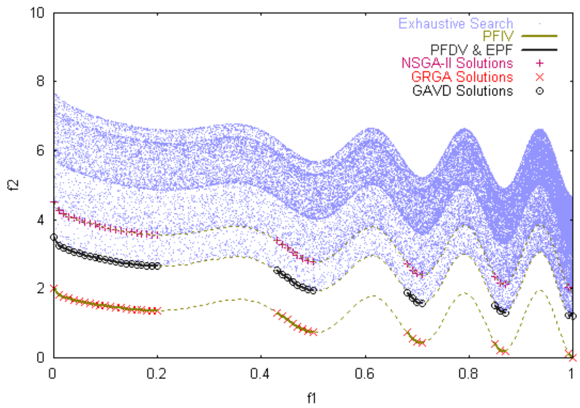


Figure 7: GAVD Performance in Problem-2 (PFIV: Pareto Front for Independent Variables, PFDV: Pareto Front for Dependent Variables, EPF: Estimated Pareto Front)

7 FUTURE RESEARCH ACTIVITIES

The current limitations of GAVD and the corresponding future research activities are as follows.

- The performance of this algorithm in identifying the dependence among decision variables is limited by the degree of RA that it uses. Hence, in dealing with complex dependence, higher order RAs are required. This implies that the use of more sophisticated non-linear modelling tools, such as Neural Networks, have the potential of improving its performance, especially in modelling deceptive and complex non-linear functions.
- GAVD also needs to be fitted with a mechanism that can learn the dependency relationships, and update it each time a new data is added, without having to repeat the whole process.
- GAVD also needs enhancements to deal with noisy data and qualitative issues in real-life problems.

8 CONCLUSIONS

There is currently a lack of systematic research in the field of variable dependence. This paper proposes an algorithm capable of handling variable dependence in multi-objective optimisation problems. The performance of proposed algorithm is compared to that of two state-of-the-art optimisation algorithms (NSGA-II and GRGA) using two dependent-variable test problems. It is observed that the proposed algorithm GAVD enables its optimisation engine (GRGA) to handle variable dependence in optimisation problems.

Acknowledgements

The authors wish to acknowledge the support of the Engineering and Physical Sciences Research Council (EPSRC) – Grant No. GR/M 71473, Nissan Technical Centre – Europe (NTC-E) and Structural Dynamics Research Corporation (SDRC) UK.

References

Banzhaf, W., Nordin, P., Keller, R.E., and Francone, F.D. (1998). *Genetic programming: An introduction.* Morgan Kaufmann Publishers, Inc., San Francisco, California (USA).

Beasley, D., Bull, D. and Martin, R. (1993). An overview of genetic algorithms: Part 2, research topics. *University computing*, vol. 15, no. 4, 170-181.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2000). *A fast and elitist multi-objective genetic algorithm: NSGA-II.* KanGAL Report No. 200002, Kanpur Genetic Algorithms Laboratory (KanGAL), Indian Institute of Technology (IIT), Kanpur (India).

Draper, N.R. and Smith, H. (1998). *Applied regression analysis.* John Wiley and Sons, Inc., New York (USA).

Evans, J.R. and Olson, D.L. (2000). *Statistics data analysis, and decision modelling.* Prentice Hall (USA).

Frees, E.W. (1996). *Data analysis using regression models: The business perspective.* Prentice Hall (USA).

Gershenfeld, N. (1999). *The nature of mathematical modelling,* Cambridge University Press, Cambridge (UK).

Harik, G.R. (1997). *Learning gene linkage to efficiently solve problems of bounded difficulty using genetic algorithms.* PhD. Thesis, Computer Science and Engineering, University of Michigan (USA).

Hertz, J.A., Krogh, A.S., and Palmer, R.G. (1991). *Introduction to the theory of neural computation.* Addison-Wesley, Redwood City, CA (USA).

Larranaga, P., Etxeberria, R., Lozano, J.A., and Pena, J.M. (1999). *Optimization by learning and simulation of Bayesian and Gaussian networks.* Technical Report No. EHU-KZAA-IK-4/99, Intelligent Systems Group, Department of Computer Science and Artificial Intelligence, University of the Basque Country (Spain).

Muhlenbein, H. and Mahnig. T. (1999). FDA - A scalable evolutionary algorithm for the optimization of additively decomposed functions. *Evolutionary computation,* vol. 7, no. 4, 353-376.

Pedrycz, W. (1998). *Computational intelligence – an introduction.* CRC Press, New York (USA).

Pelikan, M., Goldberg, D.E., and Cantu-Paz, E. (1998). *Linkage problem, distribution estimation, and Bayesian networks.* IlliGAL Report No. 98013, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign (USA).

Phadke, M.S. (1989). *Quality engineering using robust design.* Prentice-Hall International Inc., London (UK).

Rao, S.S. (1996). *Engineering optimization – theory and practice.* Wiley-Interscience, USA.

Richards, W. (1998). *Natural computation.* MIT Press, Cambridge, MA (USA).

Roy, R., Tiwari, A., Munaux, O. and Jared. G. (2000). Real-life engineering design optimisation: features and techniques. In: Martikainen, J. and Tanskanen, J. (eds.). *CDROM Proceedings of the 5th online world conference on soft computing in industrial applications (WSC5) –* ISBN 951-22-5205-8, IEEE (Finland).

Taguchi, G. (1987). *System of experimental design.* Clausing, D. (ed.), UNIPUB/Kraus International Publications, vol. 1 and 2, New York (USA).

Tiwari, A., Roy, R., Jared, G. and Munaux, O. (2001). Interaction and multi-objective optimisation. In: Spector, L., Goodman, E., Wu, A., Langdon, W.B., Voigt, H.-M., Gen, M., Sen, S., Dorigo, M., Pezeshk, S., Garzon, M. and Burke, E. (eds.). *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001),* 671-678, Morgan Kaufmann Publishers, San Francisco (USA).