Research Journal of Applied Sciences, Engineering and Technology 6(24): 4621-4629, 2013
DOI:10.19026/rjaset.6.3477
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: February 02, 2013 Accepted: March 02, 2013 Published: December 25, 2013

Research Article

Multi-Objective Optimization of a Complex System using GPSIA+DS

[1]Okafor Ekene Gabriel, [2]Sun Youchao and [3]Uhuegho Osaretin Kole
[1]National Research Institute for Chemical Technology, Nigeria
[2]Nanjing University of Aeronautics and Astronautics, Nanjing, China
[3]Nigerian College of Aviation Technology, Nigeria

Abstract: In this study, an efficient biologically inspired constrained multi-objective optimization algorithm called Genetic Pareto Set Identification Algorithm plus Different Sex (GPSIA+DS) was developed. A complex system comprising of mixed configuration, k-out-of-n and redundant subsystem was used to validate GPSIA+DS in comparison with Genetic Pareto Set Identification Algorithm (GPSIA) and Fast Non–Dominated Sorting Genetic Algorithm plus Constrain Domination (NSGA-II+CD). The optimization strategy based on Genetic Pareto Set Identification Algorithm (GPSIA) only considered feasible solutions. That is, solutions, which satisfies all constraints conditions. Infeasible solutions could facilitate faster convergence to the true Pareto front. GPSIA+DS, which considers feasible and infeasible solutions in it optimization strategy was shown to outperform GPSIA and NSGA-II for the test problem considered.

Keywords: Complex system, constrain, Genetic Algorithm (GA), Multi-Objective Optimization (MOO), Pareto set

INTRODUCTION

According to Kramer (2010) many continuous optimization problems in practice are subject to constraints. Constraints can make an easy problem hard and a hard problem even harder. Surprisingly, in the past only little research efforts have been devoted to the development of efficient and effective constraint-handling techniques-in contrast to the energy invested in the development of new methods for unconstrained optimization. The performance of a constrained multi-objective optimization algorithm is determined both by the search algorithm and the constraint-handling techniques used.

Over the past two decades, there has been an increasing interest in the application of heuristics and meta-heuristics optimization techniques, especially those population-based heuristics, to multi-objective optimization problems. This is indicated by a considerable volume of publications and research activities in this field see Hansong (2006), Okafor and Sun (2012) and Bean and Hadj-Alouane (1992). However, only a small portion of these studies are focused on constrained multi-objective optimization techniques.

Okafor and Sun (2012) developed an efficient GA-based multi-objective optimization algorithm called GPSIA to tackle multi-objective optimization problem of a series-parallel engineering system. The optimization strategy in their study only considered feasible solutions. That is, solutions, which satisfies all constraints conditions.

In review of the previous studies on constrained heuristics multi-objective optimization, it is observed that there is a noticeable lack of research efforts on either developing new constraint-handling techniques especially for multi-objective optimization, or applying advance single-objective constraint-handling techniques to multi-objective optimization. Furthermore, Hansong (2006) stated that there exist only a few systematic studies in the literature were comparative studies were performed to compare the performance of different constrained multi-objective optimization algorithms.

Hence, the aim of this study is to develop a new biologically inspired constrained multi-objective optimization algorithm called Genetic Pareto Set Identification Algorithm plus Different Sex (GPSIA+DS), which considers the feasible and infeasible solution in it optimization strategy. GPSIA+DS is based on GPSIA that was proposed previously by Okafor and Sun (2012). Furthermore, GPSIA+DS will be applied to a bi-objective optimization problem of a complex engineering system in comparison with GPSIA and NSGA-II.

CONSTRAINED MULTI-OBJECTIVE OPTIMIZATION PROBLEM

Generally, constrained multi-objective programming problem is defined as follows: find a solution $X = (X_1, X_2, \ldots, X_n)^T$ in the n-dimensional

Corresponding Author: Okafor Ekene Gabriel, National Research Institute for Chemical Technology, Nigeria

solution space X^* that minimizes the objective functions $f_m(X)$. Mathematically, this is shown below:

Minimize $H_m(X)\ m = 1,2, \dots M$

Subject to:
b inequality constraints $g_i\,(X)\ \geq 0, i = 1, 2, \dots, b$
q equality constraints $h_i\,(X) =\ 0, i = 1, 2, \dots, q$

A feasible solution $X \in X^*$ satisfies all b inequality and q equality constraints. All the Pareto-optimal solutions are subset of the feasible solutions. $H_m(X)$ is the fitness function.

EXISTING CONSTRAINT HANDLING TECHNIQUES IN OPTIMIZATION

Penalty function: This is the most common constraint-handling technique used in multi-objective optimization. In this method, the fitness function $H_m(X)$ for each solution is constructed as the sum of the objective function $f_m(X)$ and a penalty term, which depends on the constraint violations. This is given by Eq. (1):

$$H_m(X) = f_m(X) + (Ct)^{\propto} G(X) \quad (1)$$

At generation *t*, parameter *C* and are user defined; typical setting are C = 0.5, $\propto = 1$, *or* 2. $G(X)$ is a measure for the constraint violation. A frequent definition is shown in Eq. (2):

$$G(X) = \sum_{i=1}^{b} max[0, g_i(X)]^{\beta} + \sum_{i=1}^{q} |h_i(X)|^{\gamma} \quad (2)$$

where, $\beta \geq 1$ and $\gamma \geq 1$. Generally, techniques based on this approach decrease the fitness of infeasible solutions by taking the number of infeasible constraints or the distance to feasibility into account see Fiacco and McCormick (1964), Homaifar *et al*. (1994), Kuri-Morales and Quezade (1998) and Kramer and Schwefel (2006). In the death penalty approach by Kramer and Schwefel (2006), infeasible solutions are rejected and new solutions are created until enough feasible ones exist. In the co-evolutionary penalty function approach by Coello (2002), the penalty factors of an inner evolutionary algorithm are adapted by an outer evolutionary algorithm.

Repair algorithm: According to Belur (1997) and Coello (2002) these approaches either replace infeasible solutions or only use the repaired solutions for evaluation of their infeasible pendants. This class of algorithms can also be seen as local search methods that reduce the constraint violation. The repair algorithm generates a feasible solution from an infeasible one. In general, defining a repair algorithm can be as complex as solving the problem itself.

Feasibility preserving representation and operators: These techniques force candidate solutions to be feasible see Paredis (1994). A famous example is the GENOCOP algorithm by Michalewicz and Fogel (2000) that reduces the problem to convex search spaces and linear constraints. A predator-prey approach to handle constraints is proposed by Paredis (1994) using two separate populations. A comprehensive overview to decoder-based constraint handling techniques is given by Coello (2002).

Decoder functions: According to Coello (2002), Paredis (1994) and Michalewicz and Fogel (2000) decoder functions build up a relationship between the constrained solution space and an artificial solution space easier to handle. They map a genotype into a feasible phenotype. By this means even quite different genotypes may be mapped onto the same phenotype. Eiben and Smith (2003) define decoders as a class of mappings from the genotype space $\mathcal{S}'$ to the feasible regions $\mathcal{F}$ of the solution space $X^*$with the following properties: every $X \in \mathcal{S}'$ must map to a single solution $\in \mathcal{F}$, every solution $X \in \mathcal{F}$ must have at least one representation $\in \mathcal{S}'$ and every $X \in \mathcal{F}$ must have the same number of representations in $\mathcal{S}'$ (this need not be one).

Multi-Objective optimization techniques: Coello (1999) stated that these are based on the idea of handling each constraint as an objective function. Under this assumption many multi-objective optimization methods can be applied. These techniques were used by Surry *et al*. (1995), Jimenez and Verdegay (1999). In the behavioral memory method by Schoenauer and Xanthakis (1993) the Evolution Algorithm (EA) concentrates on minimizing the constraint violation of each constraint in a certain order and optimizing the objective function in the last step. Montes and Coello (2005) introduced a technique based on a multi-member Evolutionary Strategy (ES) with a feasibility comparison mechanism. Mezura-Montes (2009) gave a survey of constraint-handling methods for evolutionary algorithms.

PROPOSED BIOLOGICALLY INSPIRED CONSTRAIN HANDLING TECHNIQUE

Detailed description of GPSIA was presented in Okafor and Sun (2012). GPSIA simply combined heuristic approach and genetic algorithm in order to filter good solution in three stages. The first stage selects best solutions based on individual objective function fitness assessment. Superior or improved solutions from stage I are transferred to stage II. At stage II, weighted sum approach based on randomly generated weight was implemented genetically. In stage III, heuristic based Pareto filter technique is applied to the successful solutions generated from stage II, in order to generate the Pareto set.

In this study GPSIA plus Different Sex (DS) approach is proposed. Every individual in the GPSIA+DS approach is assigned to a feature called its sex. Similar to nature, an individual sex is identified

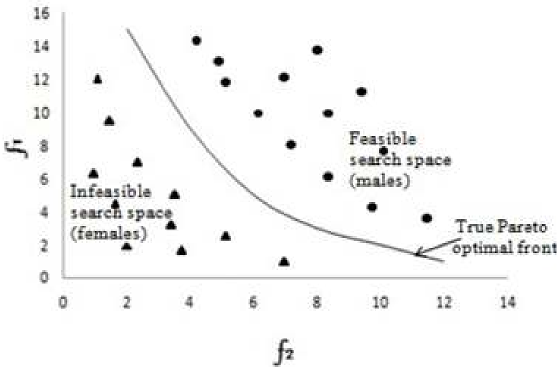

Fig. 1: Shows the feasible solution, infeasible solutions and the true POF

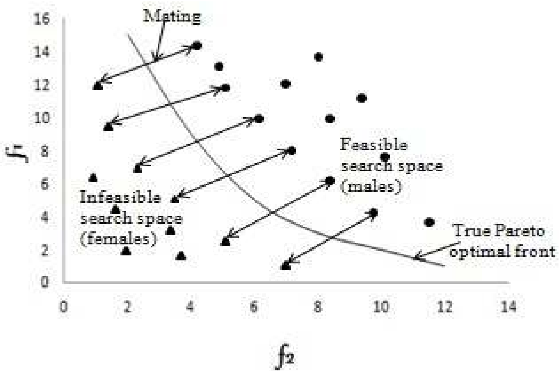

Fig. 2: Shows mating between the feasible and infeasible solutions

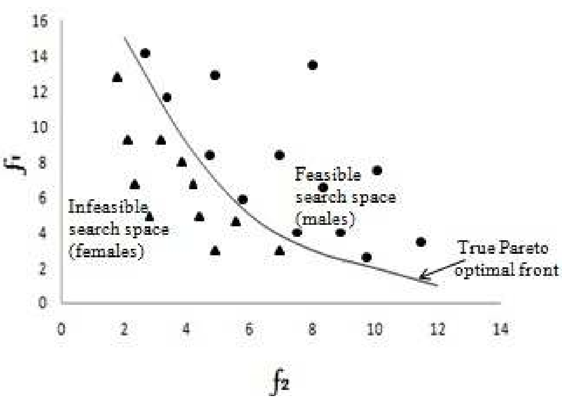

Fig. 3: Shows the feasible and infeasible solutions converging to the true POF

based on satisfaction of constrain conditions. Individuals with sex *fm* do not satisfy constrain conditions. Hence, are referred to as females. Individuals with sex *ml* satisfies the constrain conditions and are referred to as males. The GPSIA+DS reproduction proposed is based on the following rules.

- Each selected parent of every generation is allowed to mate once with a randomly selected partner
- Mating is only possible between parent of different sex (female and male)
- Parents are reintroduce into their respective population as soon as mating is completed
- Offspring sex are identified and introduced into population of same sex
- Two offspring are produced from every mating.

The driving principle of this technique is illustrated in Fig. 1, 2 and 3.

Figure 1 shows the feasible (males) and infeasible (females) search space and the true Pareto Optimal Front (POF). The optimum of the unconstrained objective function lies beyond the boundary in the infeasible search space. In the so-called GPSIA+DS female parents (μ_{fm}) are selected out of female population (P_t^o) with sex fm, whereas male parents (μ_{ml}) are selected out of male population (P_t^c) with sex ml. The individuals with sex fm were selected according to the objective function, they tend to lie finally in the infeasible search space (black triangle) whereas the ml sex individuals are selected by the fulfillment of all constraints, hence fall in the feasible search space (black circles). The measurement *G* for the fulfillment of constraints has already been defined in Eq. (2). There are other variants of *G* measurement. In this study *G* is measured according to Eq. (3):

$$G(x) = max\{[\sum_{l=1}^{T_{nct}} h_{lj}(w_{lj}) - W^a]; 0\} \qquad (3)$$

By means of intermediate recombination (crossover) between parents of different sex, as shown in Fig. 2, all individuals get closer to the optimum of the problem, but still are found on opposite sides of the boundaries between the feasible and infeasible search space as shown in Fig. 3.

Test problem: The constrained multi-objective test problem in this study is a complex system structure shown in Fig. 4. The complex system consists of four subsystems. Subsystem 1 and 3 are mixed configuration, subsystem 2 is a *k-out-of-n* and subsystem 4 is a redundant system.

Subsystem 4 could be designed adopting any of the three redundancy options shown in Table 1. The complex system consists of 17 components and 14 component types represented by blocks numbered 1 to 14 as shown in Fig. 4.

System reliability $R_s(\lambda)$ and system cost C_s are the design objectives subject to the system weight constrain. The optimization formulation for the complex system is given by Eq. (4):

Maximize

$$R_s(\lambda) = \prod_{i=1}^{d} R_{si}$$

Table 1: Subsystem 4 possible redundancy strategy

Options	Redundancy strategy
1	Active Redundancy (AR)
2	Cold Standby with a Perfect Switch System (CSPSS)
3	Cold Standby with an Imperfect Switching System (CSIPSS)

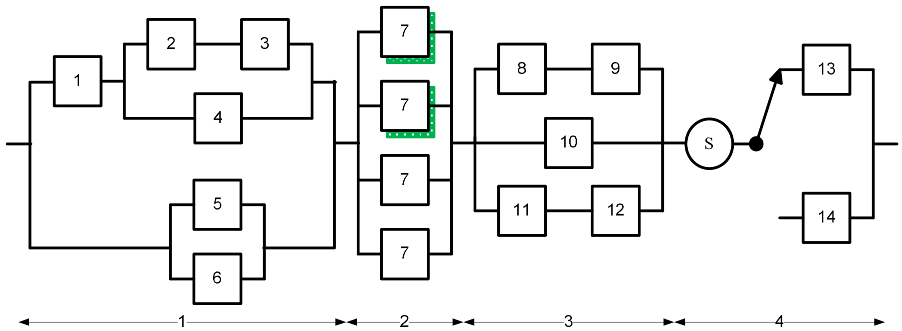

Fig. 4: Complex system

Minimize

$$C_s = \sum_{l=1}^{T_{nct}} h_{lj}.C_{lj}$$

Subject to: $\sum_{l=1}^{T_{nct}} h_{lj}.w_{lj} \leq W^a, 1 \leq j \leq 4$ (4)

where: $d = 4$, for the test problem considered in this study:

$$R_{s1} = r_{5j} + r_{6j} - r_{5j}r_{6j} + r_{1j}r_{2j}r_{3j} - r_{1j}r_{2j}r_{3j}r_{5j} - r_{1j}r_{2j}r_{3}r_{6j} + r_{1j}r_{2j}r_{3j}r_{5j}r_{6j} + r_{1j}r_{3j} - r_{1j}r_{4j}r_{5j} - r_{1j}r_{4j}r_{6j} + r_{1j}r_{4j}r_{5j}r_{6j} - r_{1j}r_{2j}r_{3j}r_{4j} + r_{1j}r_{2j}r_{3j}r_{4j}r_{5j} + r_{1j}r_{2j}r_{3j}r_{4j}r_{6j} - r_{1j}r_{2j}r_{3j}r_{4j}r_{5j}r_{6j}$$

$$R_{s2} = \sum_{i=k}^{4} C_4^k r_{7j}^k \left(1 - r_{7j}\right)^{4-k} k = 2,3,4$$

$$R_{s3} = r_{10j} + r_{8j}r_{9j} + r_{11j}r_{12j} - r_{8j}r_{9j}r_{10j} - r_{10j}r_{11j}r_{12j} - r_{8j}r_{9j}r_{11j}r_{12j} + r_{8j}r_{9j}r_{10j}r_{11j}r_{12j}$$

$$R_{s4} = r_{13j} + r_{14j} - r_{13j}\, r_{14j}\ for\ option = 1, = r_{13j} + \int_0^t f_{13j}(\tau)\, r_{14j}(t-\tau)d\tau\ for\ option = 2 = r_{13j} + \int_0^t f_{13j}(\tau)\rho_o r_{14j}(t-\tau)d\tau\ for\ option = 3$$

where T_{nct} is the total number of components types used in the system, $T_{nct} = 14$, $l = 1, 2, \ldots, T_{nct}$, h_{lj} is the repetition of option j for component type l. C_{lj} and w_{lj} is the cost and weight of option j and component type l. R_{s1}, R_{s2}, R_{s3}, R_{s4} are the reliabilities of subsystem 1, 2, 3 and 4 respectively. r_{lj} is the reliability of option j for component type l.

In Okafor and Sun (2012), weighted approach was implemented. Similar approach is implemented in this study. Given a set of weights (ω_1, ω_2) and choosing random values $\omega_1 > 0$ and $\omega_2 > 0$ according to the preference of the decision maker, the optimization problem could be stated as shown in Eq. (5):

$$Minimize\ Z = \omega_1(1 - R_s^*(\lambda)) + \omega_2 C_s^*$$

$$Subject\ to: \begin{cases} \sum_{l=1}^{T_{nct}} h_{lj}.w_{lj} \leq W^a, \\ \mathcal{L}_j \leq j \leq \mathcal{U}_j \end{cases} l = 1, \ldots, T_{nct} \quad (5)$$

where, $\mathcal{L}_j$ denote the lower bound of the available system components options and $\mathcal{U}_j$ denote the upper bound of the same variables. $\mathcal{L}_j = 1$ and $\mathcal{U}_j = 4$ was used in the study. $R_s^*(\lambda)$ and C_s^* are the normalized form of $R_s(\lambda)$ and C_s respectively. The complexity of the test problem is tunable through assigning different values to parameters shown in Table 2, changing the k value of subsystem 2, changing the redundancy strategy and system configuration.

Constrained GPSIA: GPSIA+DS: To deal with constrained multi-objective optimization problems, we proposed a biologically constrained GPSIA named GPSIA+DS (DS represent different sex) by applying the principle of different sex approach to the GPSIA published by Okafor and Sun (2012).

Solution encoding: The solution encoding is designed such that, components can be selected only in one combination amongst the j available options. The solution is therefore, encoded into a chromosome constituted by (2, T_{nct}) matrix. The complex system comprises of four subsystems. The number of columns within a system represents the number of components within that subsystem. Therefore components in subsystems 1, 2, 3 and 4 are represented by 6, 1, 5 and 2 columns respectively. The first row represents the component option selected, while the second row represents subsystems design options. Subsystems 1 and 3 are assumed to have fixed design configuration represented by zeros.

Table 2: Failure, cost and weight parameter setting

	option1 (j = 1)			option2 (j = 2)			option3 (j = 3)			option4 (j = 4)		
l	λ_{lj}	C_{lj}	w_{lj}	λ_{lj}	C_{lj}	w_{lj}	λ_{lj}	C_{lj}	w_{lj}	λ_{lj}	C_{lj}	w_{lj}
1	1.0452e5	1	6	4.1430e6	3	8	8.0256e6	2	4	1.0864e6	4	10
2	1.2511e6	2	16	2.5695e5	1	6	1.1872e6	4	18	1.4945e6	2	10
3	4.9252e6	5	14	9.1267e6	2	10	6.4595e6	3	12	1.8747e5	2	8
4	4.7181e6	4	10	1.3816e5	2	12	4.9252e6	3	8	1.1573e6	7	8
5	8.3667e6	4	8	8.8804e6	4	6	1.0094e6	5	19	1.8535e5	2	8
6	2.9566e6	3	10	2.1618e5	1	6	1.2459e6	6	15	1.9653e5	2	8
7	1.0511e5	8	18	1.1442e5	5	16	1.6806e5	3	18	1.3776e5	4	8
8	1.3220e6	3	8	1.3227e5	2	14	1.0511e6	6	12	1.1352e6	4	8
9	1.2637e5	3	16	1.3264e5	2	18	1.1195e6	5	20	2.3447e5	1	16
10	4.8827e6	8	20	5.8556e6	4	10	6.2608e6	4	12	1.8428e5	3	16
11	1.1908e5	2	10	1.5103e6	7	12	2.2328e6	5	12	6.6254e6	3	16
12	4.1590e6	6	14	4.7345e6	4	10	5.1707e6	2	12	1.6714e5	1	14
13	2.3779e5	3	10	2.3270e5	3	10	1.4146e6	6	15	2.7050e5	2	8
14	6.9815e6	4	12	1.7683e5	3	10	1.2756e6	7	17	3.8340e6	5	18

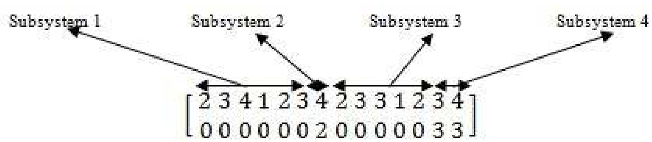

Fig. 5: Solution encoding

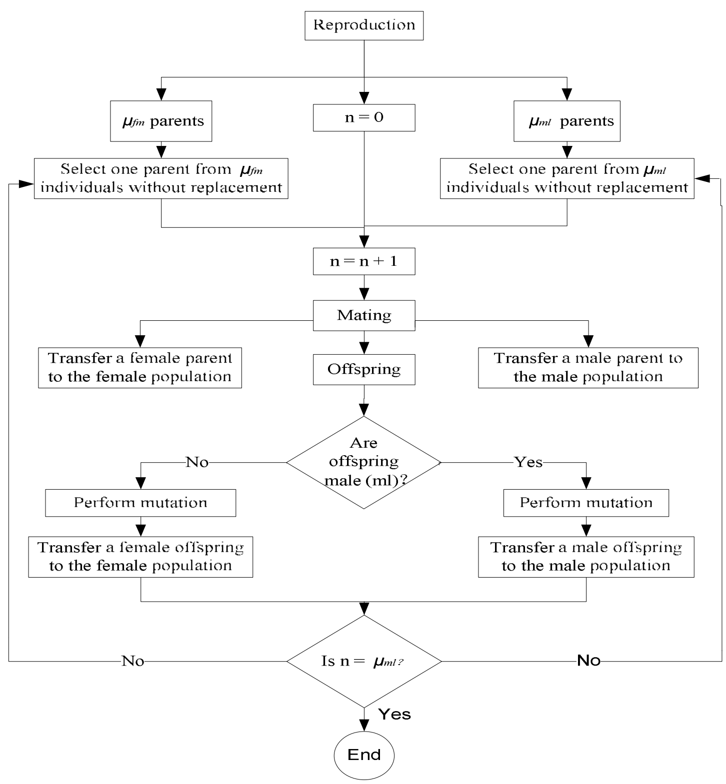

Fig. 6: GPSIA+DS reproduction strategy

$$\text{Parent A}\begin{bmatrix}2\,3\,4\,1\,2\,3\,4\,1\,3\,3\,1\,2\,3\,4\\0\,0\,0\,0\,0\,0\,2\,0\,0\,0\,0\,0\,3\,3\end{bmatrix}\quad\begin{bmatrix}2\,3\,4\,2\,2\,3\,1\,2\,3\,1\,2\,2\,3\,4\\0\,0\,0\,0\,0\,0\,2\,0\,0\,0\,0\,0\,1\,1\end{bmatrix}\text{Offspring A}$$

$$\begin{bmatrix}1\,0\,0\,1\,0\,0\,1\,1\,0\,1\,1\,0\,0\,0\\0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\end{bmatrix}$$

$$\text{Parent B}\begin{bmatrix}2\,1\,3\,2\,2\,3\,1\,2\,4\,1\,2\,2\,4\,4\\0\,0\,0\,0\,0\,0\,3\,0\,0\,0\,0\,0\,1\,1\end{bmatrix}\quad\begin{bmatrix}2\,3\,4\,1\,2\,3\,4\,1\,4\,3\,1\,2\,4\,4\\0\,0\,0\,0\,0\,0\,3\,0\,0\,0\,0\,0\,3\,3\end{bmatrix}\text{Offspring B}$$

Fig. 7: Illustration of the complex system crossover

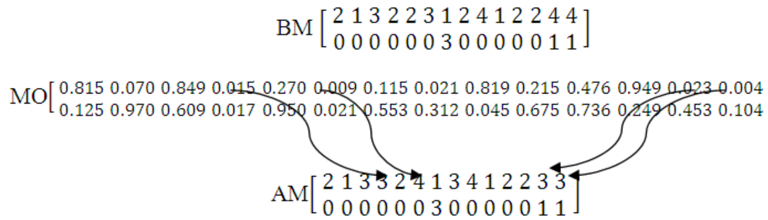

Fig. 8: An illustration of the complex system mutation strategy

Subsystem 2 is a *k-out-of-n* system. *k* can take values 2, 3 or 4, while *n* is set to 4. Subsystem 4 can be designed adopting any of the three redundancy strategies. They are AR, CSPSS, CSIPSS, corresponding to values 1, 2, 3 in the chromosome respectively. An example of the solution encoding is shown in Fig. 5. This figure represents a possible solution with option 1 of component type 4 selected in subsystem 1, option 4 of component type 7 selected in subsystem 2 and k equals to 2. Option 2 of component type 12 selected in subsystem 3 and option 4 of component type 14, with CSIPSS redundancy strategy selected in subsystem 4.

Initial population: A population size of $P = P_c + P_o$ chromosomes was used in this study, where $P_c = 10$ and $P_o = 10$.

Crossover: Male parents (μ_{ml}) were selected from population P_t^c based on the fitness value shown in Eq. (6) and the selection probability $S_\rho^c(x)$, while female parents μ_{fm} parents were selected from the population P_t^o based on the dynamic penalty function shown in Eq. (7) and the selection probability $S_\rho^o(y)$. Since it is a minimization problem, parents with lower fitness value in their population are better and has higher chances of been selected. $\mu_{fm} = \mu_{ml} = 5$ was used in this work. Ten offspring are produced from the reproduction between parents in every generation. Offspring sex are identified and combined with the existing solutions of same sex. Solutions of size P_c with the best fitness values were selected from population P_t^c and transfer to stage III. A crossover probability of 0.8 was used in this work as the breeding operator. Offspring are produced via a uniform crossover breeding operator. A binary window of same size with that of the chromosomes is generated based on the breeding operator as described in Okafor and Sun (2012). Figure 6 illustrates this reproduction strategy using flow diagram. An illustration of the complex system crossover is shown in Fig. 7.

Mutation: The mutation operator executes random alterations to the selected solutions based on a simple mutation condition. The value within the solution matrix is randomly modified at a mutation rate of 0.05. Random mutation operator of same size with the chromosome was generated with a MATLAB function rand (2, 14). Mutation occurs at all points in the offspring corresponding to same point in the random mutation operator with values less or equal to mutation rate. Chromosomes with component option *j* values of 1 to 3 at all points in which mutation can occur are converted to *j*+1. Otherwise are convert to *j*-1. Subsystem 2, with *k* value of 2 or 3 is converted to *k*+1. Else it is converted to *k*-1. The redundancy option of subsystem 4 with values 1 or 2 is converted to 2 or 3 respectively. Otherwise it is converted to 2. Figure 8 illustrates a possible mutation solution for the complex system. Mutation is ignored for all zero points.

Where, BM, MO and AM represents before mutation, mutation operator and after mutation respectively.

Pareto set selection: The fitness values of successful solution set from stage II points were computed using Eq. (6) and the current frontier points were be identified based on their fitness values:

$$H_{fi} = \left[1 - \max_{l \neq i}\left(\min\left(f_{as1}^l - f_{as1}^i, \dots, f_{asm}^l - f_{asm}^i\right)\right)\right] \quad (6)$$

where H_{fi} denotes the fitness value of the i^{th} solution (chromosome) from the set of solutions transferred to stage III; f_{ask}^i is the scaled k^{th} objective function value of the i^{th} solution (chromosome), $k = 1, \dots, m$. The maximum *(max)* in Eq. (6) is over all other solutions $l \neq i$ in the set and the minimum *(min)* is over all the objectives. The objectives, $f_{as1}, f_{as2}, \dots, f_{asm}$ in the Eq. (6) are scaled to take values between 0 and 1. Scaling is achieved using Eq. (7):

$$f_{as1}^i = \frac{f_{bs1,i} - \min f_{bs1,}}{\max f_{bs1,} - \min f_{bs1,}} \quad (7)$$

where $f_{bs1,i}$ denotes unscaled valued of the first objective for i^{th} solution (chromosome); $\max f_{bs1,}$ denotes the maximum unscaled valued of the first objective of all solutions in the population; and $\min f_{bs1,}$ denotes the minimum unscaled valued of the first objective of all solutions in the population. In case that an objective function is a constant, the scaled objective function f_{as1}^{i} is taken as 1 in this study.

GPSIA+DS Algorithm:

Stage I: Create subpopulation of size N_s^c and N_s^o where $N_s^c = {}^{P_c}/_{m}$ and $N_s^o = {}^{P_o}/_{m}$

Step 1: Start with a random initial population P_t. Set $t = 0$,

where $P_t = P_t^c + P_t^o$

Step 2: If the stopping criterion is satisfied, return P_s,

Step 3: For the objectives, $k = 1, \dots, m$, perform the following steps:

Step 3.1: For $i = 1, \dots, P_c$ and $j = 1, \dots, P_o$ assign fitness value $f_{ki}^c = f_k^c(x_i)$ and $f_{kj}^o = f_k^o(y_i)$ to the i^{th} *and* j^{th} solution respectively.

Step 3.2: Based on the fitness values assigned in step 3.1 select N_s^c best solutions between 1 and ${P_c}^{th}$ solutions and N_s^o best solutions between 1 and ${P_o}^{th}$ solutions to create subpopulation P_k^c and P_k^o respectively.

Step 4: Combine all subpopulations $(P_1^c, \dots, P_k^c)$ to create P_{t+1}^c and all subpopulations $(P_1^o, \dots, P_k^o)$ to create P_{t+1}^o of size P_c and P_o respectively. Set $\text{Gen} = \text{Gen} + 1$, *set* $P_t^c = P_{t+1}^c$, $P_t^o = P_{t+1}^o$

Stage II:

Step 5: Assign a fitness value for solution $x \in P_t^c$ and $y \in P_t^o$, by performing the following steps:

Step 5.1: Calculate the random weight of each objective k *as* $\omega_k = u_k / \sum_{i=1}^{m} u_i$

Step 5.2: Calculate the fitness of all solutions contained in P_t^c and P_t^o using Eq. (8) and (9):

$$f^c(x) = \sum_{k=1}^{m} \omega_k f_k^c(x) \tag{8}$$

$$f^o(y) = \sum_{k=1}^{m} \omega_k f_k^o(y) + Ct\left(max\left(\left[\sum_{i=1}^{T} h_{kij}(w_{kij}) - W^m\right], 0\right)\right) \tag{9}$$

where $\omega_1 f_1(.) = \omega_1(1 - R_s^*(\lambda))$ and $\omega_2 f_2(.) = \omega_2 C_s^*$

Step 5.3: Calculates the selection probability of each solution $x \in P_t^c$ and $y \in P_t^o$ *as* follow:

$$S_p^c(x) = \left(f^c(x) - f_c^{min}\right) / \sum_{v \in P_t}\left(f^c(v) - f_c^{min}\right) \text{ where } f_c^{min} = \min\{f^c(x) | x \in P_t^c\}$$

$$S_p^o(y) = \left(f^o(y) - f_o^{min}\right) / \sum_{u \in P_t}\left(f^o(u) - f_o^{min}\right) \text{ where } f_o^{min} = \min\{f^o(y) | y \in P_t^o\}$$

Step 5.4: Perform crossover and mutation on the selected parents based on their selection probability

Step 5.5: Compute offspring fitness.

Step 6: For Q generated offspring, if H $(H \leq Q)$ offspring satisfies constrain condition insert H solutions into P_t^c and $Q - H$ solutions into P_t^o. Compare the fitness values of the H and $Q - H$ offspring with those of solutions existing in P_t^c and P_t^o respectively. Eliminate the weakest H and $Q - H$ solutions from P_t^c and P_t^o respectively

Step 7: If $t = 1$ randomly select z $[3 \leq z \leq P_c]$ best solutions. Else select P_c solutions

Stage III

Step 8: Compute the fitness value based on Eq. (6) and identify current frontier points.

Step 8.1: For all solutions *z* or P_c from step 7,

If t = 1

Retain all solutions along with the current frontier points

Else *if* $1 \leq H_{fi} \leq 2$

Retain solution point (current frontier points)

Else

Discard solution.

End

End

Step 9: Combining the frontier points from last step with current frontier points in the archive.

Step 9.1: Identify the new current frontier points of the combined solution points (P_{s+1})

Step 9.2: Set $P_s = P_{s+1}$

Step 10: Once termination criterion is satisfied compute the distances between successive Pareto points. Else Set $t = t + 1$ and set $P_t = P_{t+1}$ go to step2. End

where, m is the number of objective function, f_{ki} is stage I fitness value for the k^{th} objective and the i^{th} solution, $f^c(x), f^o(y)$ is stage II fitness values for solution $x \in P_t^c$, $y \in P_t^o$ respectively, N_s^c and N_s^o is subpopulation sizes for the male and female individuals respectively, P_c, P_o is the initial population size for the male and female individuals respectively, P_t is the total population at generation *t*, P_t^c, P_t^o is the male and female population at generation *t* respectively, w_i is weight of the i^{th} objective functions, f^{min} is minimum fitness value at stage II for all $f(x)/x \in P_t$, $f_k^o(y), f_k^c(x)$ is the fitness value of the k^{th} objective for female and male solutions respectively. P_s is the Pareto set. *C* is a constant. *C* = 0.5 was used.

Table 3: Quantitative comparison between GPSIA+DS with GPSIA and NSGA-II+CD

Algorithms	System	Mean	S.D.
GPSIA+DS	Complex	1.606711	1.474096
GPSIA	Complex	1.659342	1.713463
NSGA-II+CD	Complex	1.659342	1.542854

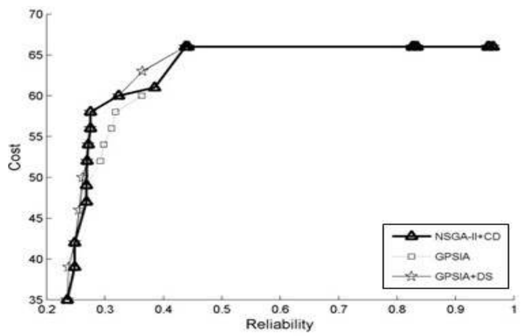

Fig. 9: Plot cost against reliability for the complex system

Stage I of GPSIA is basically same with stage I of GPSIA+DS. The main difference is that in GPSIA, step 1 to step 4 is performed for individuals in the feasible search space, while for GPSIA+DS step 1 to step 4 is performed for individuals in both the feasible and infeasible search space.

Stage II of GPSIA implemented one fitness assessment based on weighted sum, while GPSIA+DS implemented two fitness assessment strategies. The fitness of individuals in the feasible search space was computed using weighted sum approach similar to that of GPSIA developed by Okafor and Sun (2012). However, the fitness of individuals in the infeasible search space was computed using a dynamic penalty function (Eq. 9). Stage III Pareto set selection principle is based on the same techniques describe in Okafor and Sun (2012).

Comparative study:

Performance metrics: In order to ensure convergence to the true Pareto-optimal front and well-distributed set of the non-dominated solutions, many performance metrics have been proposed to evaluate the efficiency of a multi-objective optimization algorithm. Some of the commonly used once are based on spacing, spread and geometric distance. In this study spacing technique was adopted so as to enable the estimation of standard deviation associated with each algorithm compared in this study. The spacing metric is given in Eq. (10):

$$d_i^{psp} = \sqrt{\sum_{k=i}^{m}\{f(X)_k^{max} - f(X)_k^{min}\}^2} \qquad (10)$$

where d_i^{psp} is the distance between successive point in the Pareto set. $f(X)_k^{max}$ and $f(X)_k^{min}$ are the maximum and minimum values of the kth objective between successive points in the Pareto set respectively.

Experimental design: As reported before, the performance of a constrained multi-objective optimization algorithm is determined by both the multi-objective search algorithm and the constraint handling technique used. To investigate how the two components affect the performance of a constrained multi-objective optimization algorithm, a comparative study is design as follows.

In this comparative study, the proposed GPSIA+DS was compared with two multi-objective optimization algorithms. The first competitor named GPSIA, which was presented by Okafor and Sun (2012) implemented death penalty for all solution in the infeasible space. The second competitor developed by Deb (2000), called NSGA-II+CD, adopted NSGA-II search algorithm, while incorporating constrained-domination principle. The comparative studies performed in Deb (2000), demonstrated that NSGA-II+CD is a state-of-the-art algorithm in constrained multi-objective optimization.

In the experiment, the same reproduction parameter setting was used for GPSIA+DS and GPSIA. Each component has a minimum of four options to select from. For each search algorithm, a population size = 20, crossover probability = 0.8, the mutation rate = 0.05, mission time (t) = 17520h, τ = 8760h, C = 0.5 and weight constrain limit (W^a) = 206 were used.

Similarly, the comparative study between GPSIA+DS and NSGA-II+CD was done based on the same parameter setting. 50 generation was set, as the termination criteria and each algorithm was executed four times for the test problem. The simulated results are shown in Table 3 and Fig. 9. The following discussion can be made from the results.

Comparing GPSIA+DS with GPSIA: It is observed that GPSIA+DS outperformed GPSIA for the complex system considered. This shows that the proposed multi-objective version which implemented different sex constraint handling technique has a better constrain-handling ability than the death penalty, which GPSIA is based on.

Comparing GPSIA+DS with NSGA-II+CD: The simulated results revealed that GPSIA+DS out performed NSGA-II+CD on the test problems. While, GPSIA+DS showed better result on the complex system structure considered, conclusion is not reached that GPSIA+DS is better than NSGA-II+CD in all multi-objective problem. However, it is evident that GPSIA+DS is an efficient multi-objective algorithm.

CONCLUSION

In this study, an efficient biologically inspired constrained multi-objective algorithm based on the GPSIA, that was proposed by Okafor and Sun (2012) was developed. In developing the biologically constrained GPSIA, Different Sex (DS) approach, which only permits reproduction between parents of different sex (male and female), was implemented by extending it application to multi-objective problems. Solutions that satisfy the constraint conditions (feasible) were referred to as males, while those, which could not satisfy the constraint conditions (infeasible), were referred to as females. The performance of GPSIA+DS was investigated through a comparative study. In the comparative study, GPSIA+DS was compared with two constrained multi-objective algorithms (GPSIA and NSGA-II+CD). The result shows that GPSIA+DS achieved the best performance on the test problem.

REFERENCES

Bean, J.C. and A.B. Hadj-Alouane, 1992. A dual genetic algorithm for bounded integer programs. Technical Report, University of Michigan, Kalamazoo, Mich, USA.

Belur, S.V., 1997. CORE: Constrained optimization by random evolution. Proceeding of the Late Breaking Papers at the Genetic Programming 1997 Conference, Stanford University, California, Stanford Bookstore, pp: 280-286.

Coello, C.A.C., 1999. Constraint handling through a multi-objective optimization technique. Proceedings of the Genetic and Evolutionary Computation Conference Wu, A.S. (Ed.), Orlando, Fla, USA, pp: 117-118.

Coello, C.A.C., 2002. Theoretical and numerical constraint handling techniques used with evolutionary algorithms: A survey of the state of the art. Comp. Met. App. Mech. Eng., 191(11-12): 1245-1287.

Deb, K., 2000. An efficient constraint handling method for genetic algorithms. Comp. Met. App. Mech. Eng., 186: 311-338.

Eiben, A.E. and J.E. Smith, 2003. Introduction to Evolutionary Computing. Springer, Berlin, Germany.

Fiacco, A. and G. McCormick, 1964. The sequential unconstrained minimization technique for nonlinear programming: A primal-dual method. Manag. Sci., 10: 360-366.

Hansong, X., 2006. A new multi-objective optimization algorithm based on artificial immune systems and its engineering application. Ph.D. Thesis, University of Toronto.

Homaifar, A., S.H.Y. Lai and X. Qi, 1994. Constrained optimization via genetic algorithms. Simulation, 62(4): 242-254.

Jimenez, F. and J.L. Verdegay, 1999. Evolutionary techniques for constrained optimization problem. Proceedings of the 7th European Congress on Intelligent Techniques and Soft Computing (EUFIT ’99), Zimmermann, H.J. (Ed.), Mainz, Aachen, Germany.

Kramer, O., 2010. A review of constraint-handling techniques for evolution strategies. App. Comput. Intel. Sft. Comput., 2010: 1-11.

Kramer, O. and H.P. Schwefel, 2006. On three new approaches to handle constraints within evolution strategies. Nat. Comput., 5(4): 363-385.

Kuri-Morales, A. and C.V. Quezada, 1998. A universal eclectic genetic algorithm for constrained optimization. Proceeding of the 6th European Congress on Intelligent Techniques and Soft Computing (EUFIT ’98), pp: 518-522.

Mezura-Montes, E., 2009. Constraint-Handling in Evolutionary Computation. Studies in Computational Intelligence, Springer, Berlin, Germany, 198: 25-49.

Michalewicz, Z. and D.B. Fogel, 2000. How to Solve It: Modern Heuristics. Springer, Berlin, Germany.

Montes, E.M. and C.A.C. Coello, 2005. A simple multi-membered evolution strategy to solve constrained optimization problems. IEEE T. Evol. Comput., 9(1): 1-17.

Okafor, E.G. and Y.C. Sun, 2012. Multi-objective optimization of a series: Parallel system using GPSIA. Reliab. Eng. Sys. Saf., 103: 61-71.

Paredis, J., 1994. Co-evolutionary constraint satisfaction. Proceeding of the 3rd Conference on Parallel Problem Solving from Nature (PPSN ’94), Springer, Jerusalem, Israel, pp: 46-55.

Schoenauer, M. and S. Xanthakis, 1993. Constrained GA optimization. Proceeding of the 5th International Conference on Genetic Algorithm (ICGA ’93), Forrest, S. (Ed.), Morgan Kaufman, San Francisco, Calif, USA, pp: 573-580.

Surry, P.D., N.J. Radcliffe and I.D. Boyd, 1995. A multi-objective approach to constrained optimization of gas supply networks: The COMOGA method. Proceeding of the Evolutionary Computing, AISB Workshop, Fogarty, T.C. (Ed.), Lecture Notes in Computer Science Springer, Sheffield, UK, April, pp: 166-180.