

Multidisciplinary design of a small satellite launch vehicle using particle swarm optimization

Masoud Ebrahimi · Mohammad Reza Farmani ·
Jafar Roshanian

Received: 25 May 2010 / Revised: 4 April 2011 / Accepted: 19 April 2011 / Published online: 28 May 2011
© Springer-Verlag 2011

Abstract In the present paper, particle swarm optimization, a relatively new population based optimization technique, is applied to optimize the multidisciplinary design of a solid propellant launch vehicle. Propulsion, structure, aerodynamic (geometry) and three-degree of freedom trajectory simulation disciplines are used in an appropriate combination and minimum launch weight is considered as an objective function. In order to reduce the high computational cost

and improve the performance of particle swarm optimization, an enhancement technique called fitness inheritance is proposed. Firstly, the conducted experiments over a set of benchmark functions demonstrate that the proposed method can preserve the quality of solutions while decreasing the computational cost considerably. Then, a comparison of the proposed algorithm against the original version of particle swarm optimization, sequential quadratic programming, and method of centers carried out over multidisciplinary design optimization of the design problem. The obtained results show a very good performance of the enhancement technique to find the global optimum with considerable decrease in number of function evaluations.

Multidisciplinary Design Optimization of a Small Solid Launch Vehicle which is the case study of the present manuscript has been previously published and presented in the below Journals and Conferences:

- Roshanian J, Jodei J, Mirshams M, Ebrahimi R, and Mirzaei M (2010) Multilevel of Fidelity Multidisciplinary Design Optimization of Small Solid Propellant Launch Vehicle, Transaction of the Japan Society of Astronautical and Space Science, Vol. 53, No. 179.
- Jodei J, Ebrahimi M, and Roshanian J (2008) Multidisciplinary design optimization of a small solid propellant launch vehicle using system sensitivity analysis, Structural and Multidisciplinary Optimization, 38(1): 93-100.
- Jodei J, Ebrahimi M, and Roshanian J (2006) An Automated Approach to Multidisciplinary System Design Optimization of Small Solid Propellant Launch Vehicles, Proceeding of the 1st International symposium on Systems and Control in Aerospace and Astronautics, Harbin, China.

Keywords Multidisciplinary design optimization · Particle swarm optimization · Fitness inheritance · Small solid propellant extendable launch vehicle

1 Introduction

Evolutionary Algorithms (EAs) have been shown to be powerful global optimizers. These algorithms aim to utilize the attractive ability of nature to solve real world problems. According to No-Free-Lunch (NFL) theorem (Wolpert and Macreadyt 1996), there cannot exist any algorithm for solving all optimization problems that is generally superior to any competitor. The NFL theorem can be confirmed in the case of EAs versus many classical optimization methods. However, classical optimization algorithms are more efficient dealing with linear, quadratic, strongly convex, and many other special problems, EAs do not surrender so early in solving discontinuous, nondifferentiable, and noisy problems that present a real challenge to classical methods. Therefore, EAs such as Genetic Algorithm (GA) (Holland

M. Ebrahimi · M. R. Farmani (✉) · J. Roshanian
Faculty of Aerospace Engineering, K.N. Toosi University
of Technology, East Vafadar Bld., 4th Tehranpars Sq.,
16569-83911, Tehran, Iran
e-mail: mrfarmani@ieee.org, farmani60@gmail.com

M. Ebrahimi
e-mail: ebrahimim@dena.kntu.ac.ir

J. Roshanian
e-mail: roshanian@kntu.ac.ir

1992), Simulated Annealing (SA) (Kirkpatrick et al. 1983), Particle Swarm Optimization (PSO) (Kennedy and Eberhart 2001), and etc have received increasing interests recently. Though, these methods also have their own drawbacks such as high computational cost, parameter setting, constraint handling, and dealing with high dimensionality.

In recent years, some researchers have done successful works using PSO algorithm within Multidisciplinary Design Optimization (MDO) frameworks to solve optimum design problems. Venter and Sobieszcanski-Sobieski (2004) applied a PSO algorithm within a MDO framework to design a transport aircraft wing. The numerical results obviously demonstrate the PSO algorithm advantages to find optimum points dealing with the noisy and discrete variables compared with gradient based optimization algorithms. Hart and Vlahopoulos (2009) utilized a PSO algorithm at the top and discipline levels of a multi-level MDO framework. They considered a conceptual ship design problem within their integrated MDO/PSO algorithm and discussed advantages and disadvantages of their proposed methods. Peri et al. (2008) discussed some global and free-derivative optimization algorithms to solve MDO problems. A test case including a vertical fin and two numerical tools is used to demonstrate the capability of PSO for finding optimal shapes.

In the present paper, A MDO framework of a Small Solid Propellant Extendable Launch Vehicle (SSPELV), which was developed at the MDO research laboratory of K. N. Toosi University of Technology (Roshanian et al. 2010; Jodei et al. 2006, 2008), is used to minimize launch vehicle weight. Since, many engineering disciplines such as aerodynamics, propulsion, structure and trajectory are involved in the MDO of SSPELV, the design space is very complex, nonlinear, and nonconvex which may cause the gradient based methods not to work well and fall in local optimums. Therefore, we investigated the performance of a modified PSO for finding the global optimum in such a bumpy space.

However, given the high computational cost of objective function evaluations, PSO tends to become expensive. In order to improve the performance of the optimizer a new enhancement technique called Fitness Inheritance (FI) is introduced which estimates the objective function of a pre-defined portion of particles by constructing an approximate model. The proposed algorithm is compared with the original PSO, Sequential Quadratic Programming (SQP) and another gradient based optimization method called Method of Centers (MOC) (Vanderplaats 2001). This paper attempts to examine the claim that the modified PSO has the same effectiveness (finding the true global optimal solution) as the original PSO with significantly better computational efficiency (less function evaluations) dealing with the conceptual design problem of a SSPELV within a MDO structure.

2 Particle swarm optimization

Particle Swarm Optimization is inspired from the social and cognitive behavior of birds in a flock or fish in a school adapting to their environments to find a source of food. PSO leads the population of particles (swarm) towards the best area of the search space to find the global optimal solution. In PSO, a velocity vector is used to update the current position of each particle in the swarm. The velocity vector is updated based on the memory gained by each particle as well as the knowledge gained by the swarm as a whole.

The position x of a particle i at iteration $k + 1$ is updated by following equation:

$$x_i^{k+1} = x_i^k + v_i^{k+1} \Delta t \quad (1)$$

where v_i^{k+1} is corresponding velocity vector, and Δt is the time step value that is set as 1 in the present work (Shi and Eberhart 1998). The velocity vector of each particle is calculated as shown in (2):

$$v_i^{k+1} = wv_i^k + c_1r_1 \frac{(p_i^k - x_i^k)}{\Delta t} + c_2r_2 \frac{(p_g^k - x_i^k)}{\Delta t} \quad (2)$$

where v_i^k shows the velocity of particle i at iteration k , r_1 and r_2 are random numbers between 0 and 1 used to maintain the diversity of the swarm, p_i^k and p_g^k are the best position of particle i which is obtained so far (personal best), and the global best position in the swarm at iteration k (global best). Parameter c_1 represents the cognitive factor that pulls the particle to its own historical best position. Parameter c_2 represents the social factor that pushes the swarm to converge to the current globally best region. Parameter w is the inertia weight factor. This parameter is employed to control the impact of the previous history of velocities on the current one and has a critical effect on PSO's convergence behavior.

2.1 Fitness inheritance

Smith et al. (Smith et al. 1995) originally proposed a method to improve the performance of Genetic Algorithm (GA) with the aim of Fitness Inheritance (FI). The proposed method takes the average and weighted average fitness of the two parents to calculate the fitness function value of the corresponding offspring. The applied method to a simple test problem resulted in a lower computational cost without decreasing the quality of result seriously (Smith et al. 1995).

One of the proposed FI formulations in (Reyes-Sierra and Coello Coello 2005) is modified here to compute the objective function value for a new particle. This new position

in the objective space is a linear combination of objective function values for p_i^k , p_g^k , and x_i^k . Given a particle x_i^k , its personal best p_i^k , its global best p_g^k and the new particle x_i^{k+1} , the euclidean distances from x_i^{k+1} to those three positions in the search space can be presented as:

$$d_1 = d(x_i^{k+1}, x_i^k) \tag{3}$$

$$d_2 = d(x_i^{k+1}, p_i^k) \tag{4}$$

$$d_3 = d(x_i^{k+1}, p_g^k) \tag{5}$$

Following formulas calculate the objective function value for particle x_i^{k+1} :

$$r_j = \frac{\frac{1}{d_j}}{\frac{1}{d_1} + \frac{1}{d_2} + \frac{1}{d_3}} \tag{6}$$

$$f(x_i^{k+1}) = r_1 f(x_i^k) + r_2 f(p_i^k) + r_3 f(p_g^k) \tag{7}$$

where $j = 1, 2, 3$ and f is the objective function value.

Generally speaking, the last position of the particle and two attractors (personal and global bests) in the objective space are considered as parents to inherit the objective function value for the new particle. Equations (6) and (7) show that the estimated fitness is a weighted average of those of the three parents, while it coincides with that of any one parent if the positions coincide. For instance, if the new position x_i^{k+1} approximately coincides with the global best p_g^k ($d_3 \approx 0$), then $r_3 \approx 1$, and $r_1 \approx r_2 \approx 0$; and therefore $f(x_i^{k+1}) \approx f(p_g^k)$.

In order to control the quality of inheritance, a selection technique is applied here which selects the best particles in the swarm to be inherited. Zhan et al. (2007) proposed a tuning method that adaptively adjusts the value of cognitive and social factors of PSO. The method suggests an adaptive system which uses K -means algorithm (Tou and Gonzalez 1974) to distribute the population in the search space to three clusters at each iteration. This system, then, considers the relative size of the clusters containing the best and the worst particles to adjust the value of c_1 and c_2 .

In the present work, we incorporate K -means algorithm and selection rules into FI for making the inheritance more adaptive and intelligent.

K -means clustering is a partitioning method which partitions data, here position of particles in the search space, into K mutually exclusive clusters. Indeed, this algorithm finds partition in which the particles within each cluster are as close to each other as possible, and as far from particles

in other clusters as possible. By adopting the notation used in (Zhan et al. 2007), the clustering process described in the four following steps:

Step 1 Create random K cluster centers, CP_1, CP_2, \dots, CP_K for the K clusters.

Step 2 Assign particle x_i to cluster C_j , where $i \in \{1, 2, \dots, N\}$ and $j \in \{1, 2, \dots, K\}$, if and only if

$$d(x_i, CP^j) < d(x_i, CP^m), \quad 1 \leq m \leq K, \text{ and } j \neq m \tag{8}$$

where N is the number of particles in the swarm and $d(x_i, CP^j)$ is the euclidian distance between x_i and CP^j .

Step 3 Evaluate new cluster centers $CP^{1,*}, CP^{2,*}, CP^{K,*}$ as follows:

$$CP^{j,*} = \frac{1}{M_j} \sum_{x_i \in C_j} x_i, \quad 1 \leq j \leq K \tag{9}$$

where M_j is the number of particles belonging to cluster C_j .

Step 4 If $CP^{j,*} = CP^j, 1 \leq j \leq K$, then CP^1, CP^2, \dots, CP^K are chosen as the cluster centers and the clustering process will be stopped. Otherwise, assign $CP^j = CP^{j,*}$ and the process will be repeated from Step 2.

In this paper, the above process partitions swarm into three clusters ($K = 3$). At each iteration, G_B and G_W are assigned to the crowds of clusters containing the best particle and the worst particle, and, at the same time, G_{max} and G_{min} are set as the number of particles in the largest and smallest clusters. The selection states are proposed based on considering the relative size of G_B, G_W, G_{max} , and G_{min} . These states are detailed as follows:

State 1 The smallest cluster contains the best particle ($G_{min} = G_B$) and the largest cluster contains the worst particle ($G_{max} = G_W$). It is named the Initial state in which the diversity of the swarm should be protected for thorough exploration of the search space. In this state, t_k proportion of all particles called inheritance proportion will be selected from the best particles of the largest cluster to be inherited. If the population of the largest cluster is lower than t_k proportion of the swarm, the best particles of the next more populous cluster will be selected.

State 2 G_B and G_W are equal to each other and are the smallest cluster at the same time ($G_B = G_W = G_{min}$). It is named the Sub-mature state. In this state, as both the best and the worst particles

are in the smallest cluster, the swarm situation in the search space is not comprehensible easily. In such situation, it is wised to allow the swarm still explore freely. So, we choose the best particles of two other largest clusters to be inherited with an equal $t_k/2$ proportion of all particles.

State 3 G_B and G_W are equal to each other and are the largest cluster at the same time ($G_B = G_W = G_{max}$). It is named the Maturing state in which the best particle is finding the position of global optimum in the search space; however, trapping in a local optimum (prematurity) is still probable. Hence, $3t_k/4$ and $t_k/4$ proportions of all particles will be selected from the best particles of the largest and the second largest clusters respectively to be inherited.

State 4 The largest cluster contains the best particle ($G_{max} = G_B$) and the smallest cluster contains the worst particle ($G_{min} = G_W$). It is named the Matured state in which most of the particles swarmed around the best particle that is possibly the solution of the problem. Again, in this state, similar to State 1, t_k proportion of all particles will be selected from the best particles of the largest cluster to be inherited.

The four mentioned states are general forms for all other possible conditions. For example, consider this state: the largest cluster contains the best particle ($G_{max} = G_B$) and the two other clusters are equal to each other which one of them must contain the worst particle. This state can be considered a specific condition of *State 1* and undergone its rule. Other possible states have the same conditions. Since the proposed inheritance method adds only one parameter (t_k) to PSO, it does not introduce an additional design or implementation complexity. Besides, the clustering analysis is much lighter when is compared with the fitness evaluation of the being solved problem. In fact, due to the small population size of PSO, for example, 50 particles in the swarm, the clustering analysis can terminate in no more than 5 loops at each iteration.

2.2 Constraint handling

In order to use the algorithm for constrained optimization, the exterior penalty function method is applied. In this method, a penalty value is added to the objective function for each constraint violation:

$$g_q = \max \{0, \text{constraint}_q\}$$

$$\phi = f + s \sum_{q=1}^Q g_q \quad (10)$$

where Q is the number of constraints. s is called penalty parameter and the values of constraint are normalized. When using nonnormalized constraints, the following formula should be used:

$$g_q = \max \{0, \text{constraint}_q\}$$

$$\phi = f + \sum_{q=1}^Q s_q g_q \quad (11)$$

s_q values depend on the relative importance and order of magnitude of the constraints.

2.3 Fundamental steps

The following steps construct the main parts of applied PSO:

Initialization A random distribution of initial swarm is generated inside the search space and an initial set of random velocities is assigned.

Analysis The objective function is evaluated for each particle of the swarm and the values of personal best, p_i^k , and global best, p_g^k , are determined.

Updating the velocity vector The new velocity vector for each particle is calculated using (2). In this work, the inertia weight factor, w_k , is adjusted dynamically throughout the optimization process (Hart and Vlahopoulos 2009):

$$w^k = w_{max} - \left(\frac{w_{max} - w_{min}}{k_{max}} \right) k \quad (12)$$

Equation (12) contains four parameters which are defined to control the magnitude of the velocity vector during the optimization. When a bound constraint is violated, the algorithm sets the value of w_k to 0. In this way, the particle only uses its cognitive and social experiences to update its position and comes back into the feasible space.

Updating the particles' positions The positions of particles in the swarm are updated using (1).

Fitness inheritance The technique introduced in the former part is applied in this step to inherit the objective function values of t_k portion of the swarm. For the remainder particles in the swarm, true objective function values are evaluated.

Constraint handling The violation amount of each particle is evaluated and the penalty function value added to its objective function value using (10) or (11).

Convergence criterion To compare the computational cost and solution quality of the original version of PSO with the enhanced PSO, a simple convergence criterion which

Table 1 PSO parameters

Parameter	Value
Δt	1
c_1	1.4
c_2	1.4
r_1	[0,1]
r_2	[0,1]
w_{max}	0.9
w_{min}	0.1
k_{max}	200
N	Varies
t_k	Varies

is a predefined maximum iteration number, k_{max} , is used. Table 1 contains the PSO parameters used in this work.

3 Design problem

Design of complex systems such as LVs requires making appropriate compromises to achieve the optimal design among many coupled disciplines such as aerodynamics, propulsion, weight and trajectory analysis (Roshanian et al. 2010; Blair et al. 2001; Hammond 2001; Wertz 2000; Isakowitz 1995). A complex interrelation exists among mission requirements, constraints, trajectory profile, propulsion parameters, weights and loads which have to be matched by an appropriate optimization strategy. MDO involves the coordination of Multidisciplinary Design Analysis (MDA) to realize more effective solutions during the design and optimization of complex systems. It will allow system engineers to systematically explore the vast trade space in an intelligent manner and consider many more architecture during the conceptual design phase before converging on the baseline design.

In the present paper, the problem is minimizing the weight of a Small Solid Propellant Extendable Launch Vehicle (SSPELV) to launch a satellite with a known mass and inject it into a given Low Earth Orbit (LEO). There are also technological constraints such as rocket motor length, diameter, case strength and material specifications, propellant characteristics, and propellant loading density which are limited due to the availability of technologies.

3.1 Problem formulation

Several architectures can be found in the literature to solve MDO problems. Cramer et al. (1994) classify MDO methods into All-At-Once (AAO), also known as Simultaneous ANalysis and Design (SAND) (Sobieszczanski-Sobieski and Haftka 1997), Individual Discipline Feasible

(IDF) and Multiple Disciplines Feasible (MDF). Balling and Sobieszczanski-Sobieski (1996) distinguish between single-level and multilevel approaches. Single-level refers to an approach where only the system optimization problem determines the design variable values. In the multilevel case disciplinary optimizations are introduced to determine the independent discipline design variables, while the system optimizer determines the shared design variables.

MDF method (Cramer et al. 1994) can be seen as a cycle of full MDA followed by design updates, which is the most common way of solving MDO problems. The basic idea behind this method is to insert a multiple discipline analyzer between the optimizer and the disciplines. MDF has been used since nonlinear programming techniques were first applied to engineering design optimization problems and hence, it is a well established method for solving MDO problems.

In the present work, MDF architecture is applied to solve MDO problem of SSPELV. The coupling among disciplines in the conceptual design of SSPELV is shown in Fig. 1. A set of values is inserted into the variables and the disciplines are analyzed sequentially. Any analysis module, not only calculates the intermediate variable and passes them to other disciplines, but also computes function values of the equality and inequality constraints which should be satisfied.

The general layout of interdisciplinary data flow of the MDO problem is defined in Table 2. An integrated uniform design environment increases computational efficiency. All disciplinary codes have a design-oriented (batch run) version and a data dictionary allowing communication of all input and output variables without human interaction.

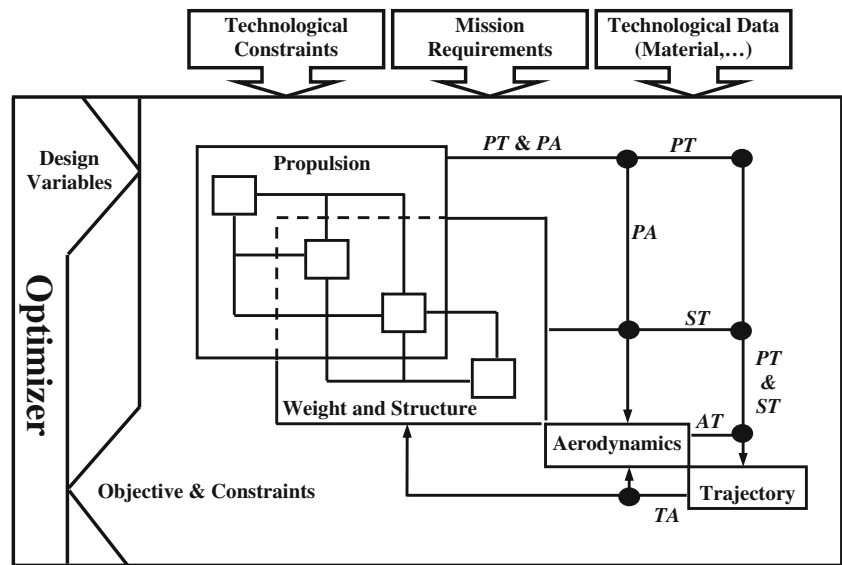
Selected design variables are classified into three types (Table 3). The variables represent geometric shapes of the vehicles (diameter of each stages, D_1 , D_2), propulsion performance (thrust and burning time of solid rocket motors (T_1 , T_2 , t_{b1} , t_{b2}) and parameters of optimized flight trajectory (maximum angle of attack during first stage maneuver $aoa_{1,max}$, final pitch angle α_f and course time t_c).

Many constraints should be satisfied in the design of LVs. This study considers several constraints as shown in Table 4; include mission (C_1 to C_3), structure (C_4 to C_6 , C_9 , C_{12}), technology (C_7 , C_8), reliability (C_{10}), and safety (C_{11}). Some of these constraints satisfy in discipline level and other must be satisfy in system level.

3.2 Mission requirements

Mission requirements consist of payload mass to be inserted into a given main mission orbit, injection accuracy, limitation on loads encountered by satellite, operational constraints such as range safety and launch azimuth. The SSPELV design reference mission for this study was to

Fig. 1 MDO conceptual design process for the SSPELV



reach a 250 km circular orbit and 34° inclination with 500 kg Orbital Maneuvering Vehicle (OMV). The launch site latitude was 34° and had eastward launch azimuth. Stage separation height should be controlled between 30–50 km. Launch site border limit causes second stage impact point to be less than 800 km. SSPELV length over diameter was limited from 8 to 13.

3.3 Analysis modules

3.3.1 Propulsion

The propulsion analysis consists of rapid solid rocket motor design code called RapSRMD. It is used for the cyclic design of solid propellant rocket motor and predicts its performance. The code consists of six design modules, three analysis modules and one simulation module. The six design modules are motor configuration design, optimum nozzle profile design, grain geometry design, case design,

insulation and igniter design. One dimensional internal ballistic simulates the motor behavior and presents thrust profile. Input parameters include motor diameter, Propellant properties, combustion thermo-chemical characteristic, thrust and burning time. Motor design algorithm scans chamber pressure and reference design height for minimum rocket motor weight. The code computes propellant weight, specific impulse, nozzle exhaust velocity, port and throat area, propellant burning area and motor dimensions.

3.3.2 Weight structure

The vehicle weight is broken down into following major subsystems: Propellant, Motor case, Nozzle, Inter-stage structure, Payload adapter, OMV, fairing, Guidance set, and Payload. Propulsion module calculates propellant weight. Other weights are estimated by statistical curve fitted Mass

Table 2 Interdisciplinary variables

Notation	Data definition	Parameters
<i>PT</i>	Propulsion to Trajectory	Propellant weight of each stage, Nozzle exit pressure and area
<i>PA</i>	Propulsion to Aerodynamics	Rocket motors length and diameter
<i>ST</i>	Structure to Trajectory	Weights of rocket motors case, nozzle, inter-stage, etc.
<i>AT</i>	Aerodynamics to Trajectory	Aerodynamic Coefficients
<i>TA</i>	Trajectory to Aerodynamics	Mach number, Angle of attack, Reynolds number, Stage number

Table 3 Design variables and their boundaries

Design variable (dimension)	Description	Minimum value	Maximum value
D_1 (m)	First stage diameter	1.3	2.5
D_2 (m)	Second stage diameter	1.3	2.5
T_1 (kN)	First stage thrust	1500	1700
T_2 (kN)	Second stage thrust	150	300
t_{b1} (sec)	First stage burning time	60	90
t_{b2} (sec)	Second stage burning time	100	170
$aoa_{1,max}$ (deg)	Maximum angle of attack during first stage maneuver	0	8
α_f (deg)	Final pitch angle	-30	15
t_c (sec)	Course time	0	30

Table 4 Constraints of SSPELV problem

Constraint	Description
C ₁	Tolerance of the terminal velocity
C ₂	Tolerance of the terminal height
C ₃	Tolerance of terminal injection angle
C ₄	Maximum dynamic pressure
C ₅	1st stage diameter limit (bigger than or equal to the 2nd stage diameter)
C ₆	Launch vehicle length-to-diameter ratio limits
C ₇	Maximum motor specific impulse
C ₈	Maximum loading of motors density
C ₉	Minimum structural fraction
C ₁₀	Separation height
C ₁₁	Range safety criteria
C ₁₂	1st stage maximum allowable angular velocity

Estimate Relationships (MES). Weight of motor case was calculated using special pressure vessels codes.

3.3.3 Aerodynamics

The external vehicle mold lines are not allowed to change. Semi empirical equations were used for aerodynamic coefficients computation. These methods developed from databases of analyses, flight tests and wind tunnel data for vehicle that were either flying or existing.

The modeling program generates tabulated aerodynamics coefficients. These tables present coefficient values relative to Mach number, Reynolds number and angle of attack. The trajectory module interpolates these data. The effect of aerodynamics shape and coefficients in system level variables has been considered.

3.3.4 Trajectory analysis

To analyze the ascent flight-path, a three degree-of-freedom trajectory analysis is performed with a program called TRAJECT. The three degree of freedom equations of motion

are numerically integrated from an initial to a terminal set of state conditions. Within the present investigation, the vehicle is treated as a point-mass. Earth rotation and oblateness are modeled, and the 1976 standard atmosphere (no winds) is used. Terminal constraints on altitude, velocity, and flight-path angle, as well as maximum flight dynamic pressure, range safety and stages separation height limits are enforced. Direct injection approach is selected, and orbital maneuver and trim are performed by OMV.

4 Comparison of results

4.1 Numerical benchmark functions

In order to compare computational cost and quality of solutions of the enhanced PSO algorithm (named PSO/FI) with PSO, a test set with four numerical benchmark functions is employed. All selected functions are well-known in the global optimization literature (Yao et al. 1999). The test set includes unimodal as well as multimodal optimization problems. f_1 and f_2 are Schwefel’s and Rosenbrock functions. They are unimodal functions and f_3 and f_4 are multimodal functions known as Rastrigin and Ackley functions. All these test functions are complex optimization problems with many local optima.

The dimensionality of the numerical functions is set to 10 to show the efficient of the enhanced method within search spaces similar to the design space of SSPELV problem that contains nine design variables. The definition, the range of search space, and also the global minimum of each function are given in Table 5.

The efficient of PSO and PSO/FI are compared by measuring the number of function calls which is one of the most commonly used metrics in the literature (Andre et al. 2001). As mentioned before, the termination criterion is meeting maximum number of iterations. In order to minimize the effect of stochastic nature of the algorithms, the results are reported over 30 runs for each test function. We also performed experiments with different inheritance proportion

Table 5 Numerical benchmark functions

Test functions	Search space	f_{min}
Schwefel : $f_1(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-10,10]^{n=10}$	0
Rosenbrock : $f_2(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$	$[-2,2]^{n=10}$	0
Rastrigin : $f_3(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$	$[-5.12,5.12]^{n=10}$	0
Ackley : $f_4(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	$[-32,32]^{n=10}$	0

Table 6 Results of PSO and FI/PSO on benchmark problems of dimensionality 10

		PSO	PSO	PSO/FI	PSO/FI	PSO/FI	PSO/FI	PSO/FI
N		50	40	30	50	50	50	50
t_k		–	–	–	0.1	0.2	0.3	0.4
f_1	Mean	1.82514e-07	2.19824e-05	1.15384e-03	3.50824e-07	1.28738e-06	1.11658e-05	3.40156e-05
	StDev	3.10849e-07	9.22063e-05	3.54571e-03	5.32506e-07	7.25902e-06	1.62103e-04	5.17801e-04
f_2	Mean	2.42572	5.74227	9.64410	2.75960	2.98741	3.57604	4.67841
	StDev	1.54861e	2.75067	4.72236	1.70125	3.63512e-01	3.03146	1.13648
f_3	Mean	7.69669	13.53181	19.01292	8.05925	8.95645	10.64828	11.146827
	StDev	3.18997	5.37865	8.65019	3.78124	4.315278	5.32610	5.67841
f_4	Mean	4.45856e-08	9.16001e-07	3.33935e-05	4.87415e-08	1.32141e-08	5.61274e-07	6.74121e-06
	StDev	4.59579e-08	9.87559e-07	5.33748e-06	7.89541e-08	1.95426e-08	5.68741e-06	1.33456e-06
NFE		10,000	8,000	6,000	9,100	8,100	7,100	6,100

Mean, standard deviation (StDev), and number of function evaluations (NFE) of 30 runs

values ($t_k = 0.1, 0.2, 0.3,$ and 0.4). These proportions of individuals indicate also the percentage by which the Number of Function Evaluations (NFE) is reduced. For example, when t_k is set to 0.5, NFE is approximately half of the original PSO ones. It is important to note that, in PSO/FI, at the first and last iterations the true values of objective functions must be evaluated for all particles. The parameters given in Table 1 are adopted for both PSO and PSO/FI algorithms.

The results for the benchmark f_1 – f_4 are shown in Table 6. Moreover, Figure 2 shows the convergence graphs

of Rastrigin function (f_3) as an example for performance comparison. For all test functions there is almost a consistent performance pattern across two algorithms: PSO/FI is better than PSO.

As it is shown in Table 6, PSO/FI approach using $t_k = 0.1$ and 0.2 is outperformed by the original PSO with 50 number of particles for all test functions. However, the differences between obtained solutions are not considerable. In other words, the optimal values found by PSO/FI indicate that our method obtained as good approximation of the true

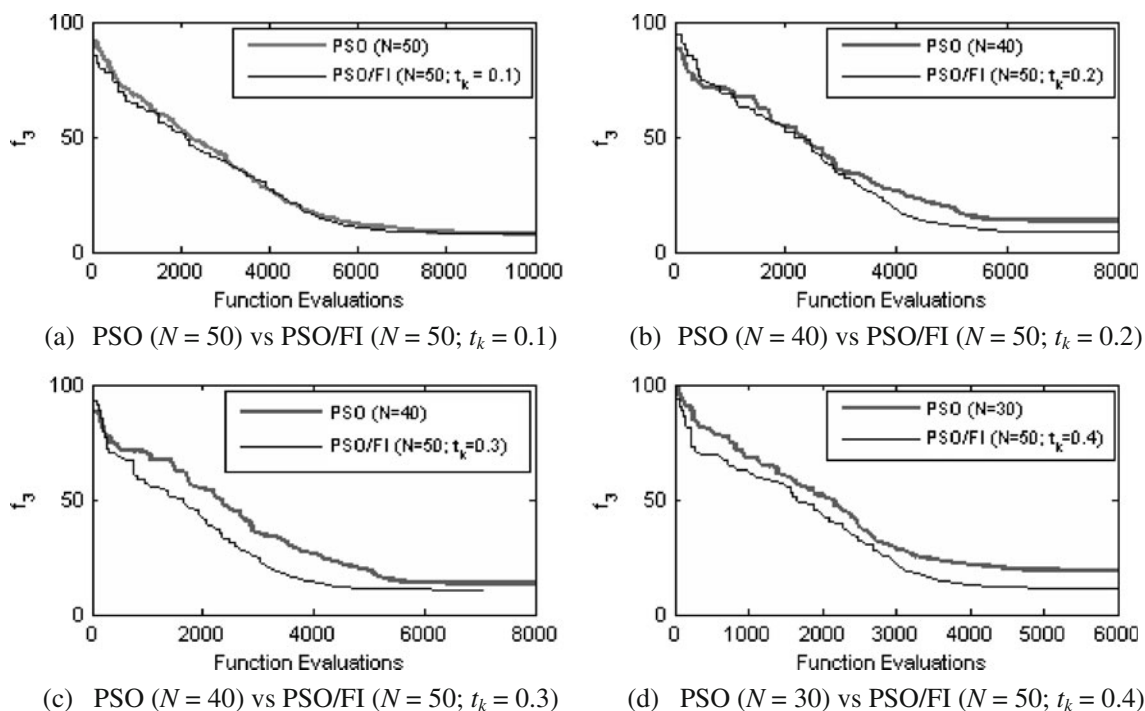


Fig. 2 Examples for performance comparison between PSO and PSO/FI for minimization problem $f_3(x) = 0$. Experiments have been repeated 30 times to plot the average values. **a** PSO ($N = 50$) vs

PSO/FI ($N = 50; t_k = 0.1$). **b** PSO ($N = 40$) vs PSO/FI ($N = 50; t_k = 0.2$). **c** PSO ($N = 40$) vs PSO/FI ($N = 50; t_k = 0.3$). **d** PSO ($N = 30$) vs PSO/FI ($N = 50; t_k = 0.4$)

Table 7 Penalty function parameters

Constraint	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}	g_{11}	g_{12}
Penalty parameter	1000	1000	10000	10000	50	50	10000	10000	10000	50	50	1000

optimal solution with lower NFE. For example, setting t_k to 0.2, PSO/FI reduced NFE by about 2,000 in comparison with PSO ($N = 50$) without having considerable effect on the quality of optimal solutions. Figure 2a and b presents the convergence history of both algorithms using above number of particles and inheritance parameters for test function f_3 .

On the other hand, setting t_k to 0.3 and 0.4, PSO/FI is clearly the best with respect to both function evaluations (efficiency) and quality of the found solutions (effectiveness). Consider the solutions obtained by PSO setting N as 40 and 30 (Table 6). It is obvious that PSO/FI had considerably better performance rather than PSO with approximately equal NFE. Figure 2b and d show the effectiveness of PSO/FI comparing with PSO for $t_k = 0.3$ and 0.4 and $N = 30$ and 40. Moreover, it is presented in Fig. 2c that PSO/FI ($t_k = 0.3$) has outperformed PSO ($N = 40$) with about less 1000 function evaluations for f_3 .

4.2 Multidisciplinary design optimization of SSPELV

In order to demonstrate proper implementation of the proposed enhancement technique within MDO structure, conceptual design of the small satellite launch vehicle is solved and results are compared to results of original version of PSO, Sequential Quadratic Programming (SQP) and Method of Centers (MOC).

We again performed experiments with the applied inheritance proportion values proposed before ($t_k = 0.1, 0.2, 0.3,$ and 0.4) and the parameters presented in Table 1 for both PSO and PSO/FI. The parameter settings for the afore-

mentioned constraint handling method (10) are summarized in Table 7. In order to embody the approximate shape of SSPELV design space, three graphs are depicted in this part. Since the design problem contains nine variables, each graph is presented by varying only two variables while other seven variables considered constant (Figs. 3, 4, and 5). Hence, the true design space must be more complex than presented here. Figures 3, 4, and 5 show the weight values as a function of final pitch angle (α_f) - course time (t_c), second stage burning time (t_{b2}) - second stage thrust (T_2), and second stage diameter (D_2) - second stage thrust (T_2), respectively. These figures demonstrate that even in these simplified conditions, involving many engineering disciplines in the MDO structure, the design space of SSPELV problem is very complex with many local optima.

Table 8 shows PSO and PSO/FI results which are representative of 30 independent trials for design of SSPELV. We also summarized the results of SQP and MOC (Vanderplaats 2001) at the two last columns of Table 8 which are found by running each method from 30 different random starting points. MOC is not a widely used optimization algorithm, but as discussed in (Vanderplaats 2001) this method is a modified version of Sequential Linear Programming (SLP) for escaping from local minima in complex optimization problems which is a serious issue for most conventional methods such as SQP. Consequently, in the present work, we applied both SQP and MOC to show their behavior dealing with our complex design problem.

MOC, also known as the method of inscribed hyperspheres (Baldur 1972), approaches the optimum point from inside the feasible region of design variables and produces a sequence of improving solutions which follow a path down the center of design variables. The basic concept of this

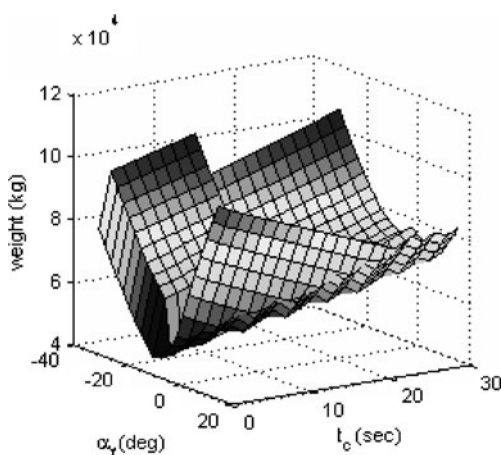


Fig. 3 Weight value variations as a function of final pitch angle (α_f) and course time (t_c)

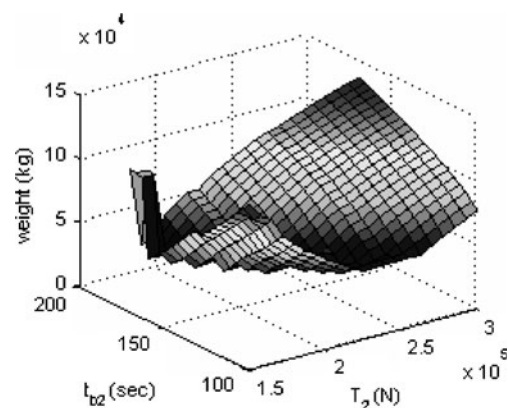


Fig. 4 Weight value variations as a function of second stage burning time (t_{b2}) and second stage thrust (T_2)

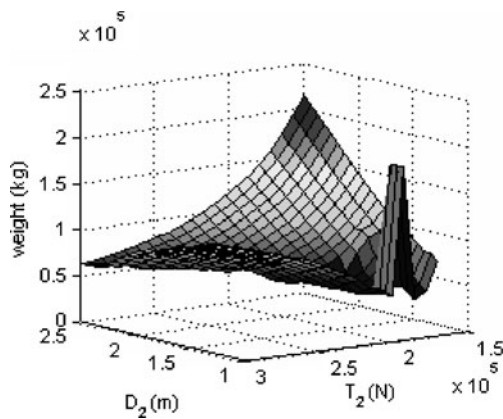


Fig. 5 Weight value variations as a function of second stage diameter (D_2) and second stage thrust (T_2)

method is to find the largest circle (hypersphere in n dimensional when n is number of design variables) which fits in a linear space of design variables and then moves to the center of that circle. This process repeats until the algorithm converges to a specified tolerance.

As we can see in Table 8, both PSO and PSO/FI provide significantly better results than SQP and MOC at a considerable higher computational cost. The optimal solutions found by all algorithms meet all constraints (presented in Table 4); however, some bound constraints including first and second stage diameter, first stage burning time, and maximum angle of attack during first stage maneuver are active in approximately all cases. It is worth to note that the bound

constraints in this work are considered regarding the baseline design variables and could be changed by designers' demands. Therefore, the obtained values of design variables which are placed on bounds are not very critical.

Based on the data presented in Table 8, MOC outperformed SQP with approximately equal NFE. By this way, the results confirm that MOC could escape from some local optimums while SQP has been trapped in them. On the other hand, the minimum weight values gained by PSO and PSO/FI prove that both MOC and SQP failed to find the global optimum.

PSO with 50 particles obtains the minimum value of weight; however, PSO/FI using $t_k = 0.1$ could preserve the quality of optimum solution perfectly well with about less than 1,000 function evaluations (Fig. 6a). In addition, it is shown that PSO/FI with $t_k = 0.2$ reduced NFE by about 2,000 (19%) in comparison with PSO ($N = 50$) with only 0.417% decrease in the quality of optimum solution.

Figure 6b and c shows the convergence history of PSO ($N = 40$) and PSO/FI ($t_k = 0.2$ and 0.3). These figures and their corresponding results in Table 8 confirm that PSO/FI shows considerably better efficiency and effectiveness. PSO/FI ($t_k = 0.2$) and PSO/FI ($t_k = 0.3$) offer 3.63% and 2.59% of decrement in the weight value with approximately equal and 11.25% lower computational cost in comparison with PSO ($N = 40$).

Finally, applying PSO/FI ($t_k = 0.4$), the quality of optimum solution has decreased about 2.79% in comparison with ($N = 50$) with a saving of 39% in function evaluations. PSO/FI ($t_k = 0.4$) also resulted in 3.56% better weight

Table 8 Results of PSO, FI/PSO, SQP, and MOC on SSPELV design problem

	PSO	PSO	PSO	PSO/FI	PSO/FI	PSO/FI	PSO/FI	SQP	MOC	
N	50	40	30	50	50	50	50	–	–	
t_k	–	–	–	0.1	0.2	0.3	0.4	–	–	
D_1 (m)	2.50	2.15	2.17	2.02	2.13	2.03	1.99	2.00	2.04	
D_2 (m)	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30	
T_1 (kN)	1,500.01	1,500.00	1,500.00	1,500.00	1,500.12	1,500.00	1,500.00	1,501.63	1,500.96	
T_2 (kN)	165.32	173.19	150.00	241.25	167.75	179.51	186.18	197.63	190.32	
t_{b1} (sec)	60.00	60.00	60.00	60.00	60.00	60.00	60.01	60.00	60.00	
t_{b2} (sec)	144.41	138.41	172.01	114.93	132.41	136.76	139.14	149.27	142.31	
$aoa_{1,max}$ (deg)	8.00	8.00	8.00	8.00	8.00	8.00	8.00	8.00	8.00	
α_f (deg)	–9.38	–11.04	–14.79	–7.29	–10.05	–11.11	–11.93	–15.01	–13.01	
t_c (sec)	26.32	28.23	29.15	26.34	27.51	27.92	28.03	29.71	29.00	
Weight (kg)	Best	44,692.21	45,415.24	45,325.06	44,756.41	44,916.41	44,976.6	45,023.4	52,347.66	49,924.14
	Worst	45,761.63	47,962.11	48,536.25	45,912.57	46,023.14	46,810.36	46,991.87	61,292.51	61,329.52
	Mean	44,901.90	46,788.77	47,856.76	44,926.84	45,089.04	45,579.16	46,154.49	54,780.91	52,350.1
	StDev	3.26	6.62	10.11	4.14	6.01	8.32	9.21	15.88	13.13
NFE	10,000	8,000	6,000	9,100	8,100	7,100	6,100	150	158	

Best, worst, mean, standard deviation (StDev), and number of function evaluations (NFE) of 30 runs. Design variables are also reported for the best weight values

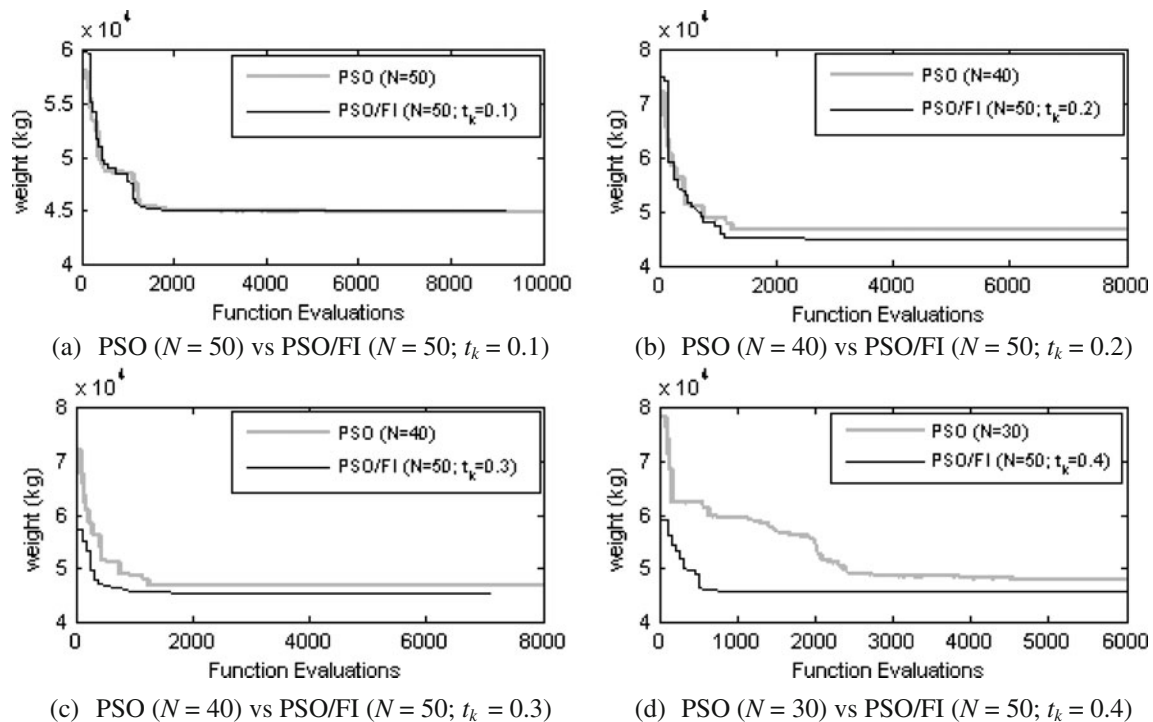


Fig. 6 Performance comparison between PSO and PSO/FI for SSPLEV design problem. Experiments have been repeated 30 times to plot the average values. **a** PSO ($N = 50$) vs PSO/FI ($N = 50; t_k = 0.1$). **b** PSO ($N = 40$) vs PSO/FI ($N = 50; t_k = 0.2$). **c** PSO ($N = 40$) vs PSO/FI ($N = 50; t_k = 0.3$). **d** PSO ($N = 30$) vs PSO/FI ($N = 50; t_k = 0.4$)

value than PSO ($N = 30$) with approximately equal NFE (Fig. 6d).

In general, in spite of the fact that both PSO and PSO/FI algorithms are more time consuming than SQP and MOC (due to their stochastic nature and the random behavior of particles), the minimum weight they generated are 14–18% lower. It is very important to note that in expensive aerospace projects such as inserting a payload mass to a defined mission orbit this amount of decreasing in gross weight is very effective and can reduce the project cost over million dollars (Hammond 2001). As a result, it can be concluded that the proposed PSO/FI technique succeeded to obtain very good approximation of true optimal solution while decreasing the computational time considerably. The found results show that PSO/FI had a very good and promising performance dealing with conceptual design of the small satellite launch vehicle within MDO structure.

5 Conclusions

In this paper, a Fitness Inheritance technique is proposed and incorporated into a Particle Swarm Optimization algorithm to minimize the weight of a solid propellant launch vehicle within the Multidisciplinary Design Optimization framework. The proposed technique was studied using different number of particles and Fitness Inheritance

proportions and compared with Particle Swarm Optimization, Sequential Quadratic Programming and Method of Centers algorithms. The obtained results show that the enhanced Particle Swarm technique has a good performance and is very promising. In fact, this method can significantly decrease the number of function evaluations without having serious bad effects on the quality of solutions. However, it is important to note that the results gained in this work are only examined and valid for four numerical test functions and the introduced design problem containing only nine to 10 decision variables. In other words, the proposed approximation technique within particle swarm optimization algorithm makes a *heuristic* method which is only designed and studied for solving the introduced problems. This method also adds a new parameter (inheritance proportion) to Particle Swarm Optimization that must be tuned based on the complexity of the search space. As a part of our future work we plan to improve and study the inheritance technique in order to solve high dimensional optimization problems with minimum decrease in quality of results.

References

- Andre J, Siarry P, Dognon T (2001) An improvement of the standard genetic algorithm fighting premature convergence in continuous optimization. *Adv Eng Softw* 32:49–60

- Baldur R (1972) Structural Optimization by Inscribed Hyperspheres. *J Eng Mech ASCE* 98(EM3):503–508
- Balling RJ, Sobieszczanski-Sobieski J (1996) Optimization of coupled systems: a critical overview of approaches. *AIAA J* 34(1):6–17
- Blair JC, Ryan RS, Schutzenhofer LA (2001) Launch vehicle design process: characterization, technical integration, and lessons learned, NASA/TP-2001-210992
- Cramer EJ, Dennis JE, Frank PD, Lewis RM, Shubin GR (1994) Problem formulation for multidisciplinary design optimization. *SIAM J Optim* 4(4):754–776
- Hammond WE (2001) Design methodologies for space transportation systems. AIAA, Reston
- Hart C, Vlahopoulos N (2009) An integrated multidisciplinary particle swarm optimization approach to conceptual ship design. *Struct Multidiscipl Optim* 41(3):481–494
- Holland JH (1992) Adaptation in natural and artificial systems. MIT, Cambridge
- Isakowitz SJ (1995) International reference guide to space launch systems, 2nd edn. AIAA, Reston
- Jodei J, Ebrahimi M, Roshanian J (2006) An automated approach to multidisciplinary system design optimization of small solid propellant launch vehicles. In: Proceeding of the 1st international symposium on systems and control in aerospace and astronautics, Harbin, China
- Jodei J, Ebrahimi M, Roshanian J (2008) Multidisciplinary design optimization of a small solid propellant launch vehicle using system sensitivity analysis. *Struct Multidiscipl Optim* 38(1):93–100
- Kennedy J, Eberhart RC (2001) Swarm intelligence. Morgan Kaufmann, San Francisco
- Kirkpatrick S, Gellatt C, Vecchi M (1983) Optimization by simulated annealing. *Science* 220(4598):671–680
- Peri D, Fasano G, Dessi D, Campana EF (2008) Global optimization algorithms in multidisciplinary design optimization. In: 2nd AIAA/ISSMO multidisciplinary analysis and optimization conference, Canada
- Reyes-Sierra M, Coello Coello CA (2005) A study of fitness and approximation techniques for multi-objective particle swarm optimization. In: Congress on evolutionary computation, IEEE Service Center
- Roshanian J, Jodei J, Mirshams M, Ebrahimi R, Mirzaei M (2010) Multilevel of fidelity multidisciplinary design optimization of small solid propellant launch vehicle. *Trans Jpn Soc Astronaut Space Sci* 53(179):73
- Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In: IEEE international conference on evolutionary computation. IEEE, Piscataway, pp 69–73
- Smith RE, Dike BA, Stegmann SA (1995) Fitness inheritance in genetic algorithms. In: SAC' 95: Proceeding of the 1995 ACM symposium on applied computing, Nashville, Tennessee, United States. ACM, New York, pp 345–350
- Sobieszczanski-Sobieski J, Haftka RT (1997) Multidisciplinary aerospace design optimization: survey of recent developments. *Struct Multidiscipl Optim* 14(1):1–23
- Tou JT, Gonzalez RC (1974) Pattern recognition principles. Addison-Wesley, Reading
- Vanderplaats GN (2001) Numerical optimization techniques for engineering design, 3rd edn. Vanderplaats Research & Development, Colorado Springs. ISBN 0-944956-01-07
- Venter G, Sobieszczanski-Sobieski J (2004) Multidisciplinary optimization of transport aircraft wing using particle swarm optimization. *J Struct Multidisc Optim* 26(1–2):121–131
- Wertz JR (2000) Economic model of reusable vs. expendable launch vehicles. In: IAF 51st international astronautical congress Rio de Janeiro, Brazil
- Wolpert DH, Macreadyt WG (1996) No free lunch theorem for search. Technical Report SFI-TR-95-02-010, Santa Fe Institute
- Yao X, Liu Y, Lin GM (1999) Evolutionary programming made faster. *IEEE Trans Evol Comput* 3(2):82–102
- Zhan Z-H, Xiao J, Zhang J, Chen W-N (2007) Adaptive control of acceleration coefficients for particle swarm optimization based on clustering analysis. In: Proceedings of the IEEE congress on evolutionary computation, Singapore, September 2007, pp 3276–3282