International Journal of
Science and Engineering Investigations vol. 4, issue 38, March 2015

ISSN: 2251-8843

Multi Target Optimization of Turbo Jet Engine with MOPSO Algorithm

Seied Hossein Azizi[1], Mahmoud Reza Andalibi[2], Siamak Rashidmardani[3]
[1,2,3]Department of Mechanical Engineering, Bandar Lengeh Branch, Islamic Azad University, Bandar Lengeh, Iran
([1]s.hossein.azizi@gmail.com, [2]andalibi.mr@hotmail.com, [3]ali_Rashidmardani@yahoo.com)

Abstract**-** In this paper, turbojet engine in ideal condition will be optimized by multi target genetic algorithm. The target functions are specific thrust (ST), specific fuel consumption (SFC) and thermal efficiency (ηt) that once will simultaneously be optimized by two by two way and the results will be revealed in the Pareto curves. For the second time these three objective functions will be optimized at the same time. At the end the findings of two by two ways will be compared with the results of three objective functions.

Design variables are considered as Mach number and total compressor pressure ratio. The significant relation between objective functions is introduced according to Pareto points. There is no doubt that these functions without using methods are not considerable.

Keywords- *Genetic Algorithm, Pareto, Multi target optimization, Crossover, Mutation, Turbojet engine.*

I. INTRODUCTION

Simulation and optimization models can provide solutions to many problems in the field of operations research. Although simulation models are based on trial and test methods followed by engineering judgment, final solutions may not be optimal.

In contrast, optimization models can yield optimal/near-optimal solutions by searching a part or an entire decision space. Different single objective optimization methods such as linear (LP), nonlinear (NLP) and dynamic programming (DP) are capable to move toward optimal solutions. However, difficulties in determining optimal solutions, especially in some discrete or nonlinear problems, as well as the curse of dimensionality in solving the large-scale problems, are disadvantages of those optimization methods.

Evolutionary algorithms are potential candidates to determine optimal/near-optimal solutions in the aforementioned problems.

In these types of algorithms, random decision variables are produced as input data for a simulation model. Output data from the simulation model can be used as input data for an optimization model. In such a process, newly-generated decision variables, based on previously calculated ones, can be improved. This process continues up to the maximum number of iterations for determining the best solution. Genetic algorithm (GA), particle swarm optimization (PSO), ant colony optimization (ACO), and simulated annealing (SA) are evolutionary algorithms that have been developed for solving optimization problems. Multi-objective problems are another type of operations research problems which include a vector of objectives instead of a single objective. The main goal of multi objective optimization techniques is to determine a set of optimal solutions, especially when objectives are conflicting.

Several computational intelligence based approaches, namely, evolutionary computation, swarm intelligence and artificial immune systems have been used for solving MOO problems. PSO and ant colony optimization methods belong to the swarm intelligence domain of computational intelligence. The population based nature of evolutionary techniques captures the different compromising solutions in the population simultaneously at each iteration. This fact has led to the considerable growth in multi-objective evolutionary algorithms (MOEA), from VEGA in [13], to the most recent techniques like NSGA-II [4], SPEA-2 [16] and PESA-II [3]. A population based swarm intelligence heuristic called Particle Swarm Optimization (PSO) was proposed in 1995 [9]. This is inspired by the flocking behavior of birds, which is very simple to simulate and has been found to be quite efficient in handling the single objective optimization (SOO) problems [5],[14]. The simplicity and efficiency of PSO motivated researchers to apply it to the MOO problems since 2002. Some of these techniques can be found in [1], [2], [6], [7], [10], [11], [12], [15].

In this study, design variables such as inlet Mach number and total compressor pressure ratio are considered. Selective multi target in ideal subsonic turbo jet included specific thrust, specific fuel consumption and thermal efficiency and with considering design variables will be optimized two by two. The results will be revealed by Pareto curves. Our goal is decreasing specific fuel consumption and increasing specific thrust and thermal efficiency.

II. TURBO JET THERMODYNAMIC MODEL

Operating fuel in turbo jet engine is air which by changing in kinetic energy in inlet comparing with outlet can create thrust.

Ideal turbojet engine equations are shown in Appendix II [8]. Inlet parameters in this cycle included flight Mach number (M_0), inlet air temperature (T_0, K), temperature coefficient (γ), heating value (h_{pr}, kj.kg^{-1}), burner exit total temperature (T_{t4},K), total compressor pressure ratio (π_c).

Outlet parameters involves specific thrust (ST, N.kg^{-1}.S^{-1}), fuel/air ratio (f), thrust specific fuel consumption (TSFC, kg.S^{-1}.N^{-1}) and thermal efficiency (η_t).

In this study h_{pr}=48000 kj.kg-1, γ=1.4, T_{t4}=1666K, T_0=216.6K. Flight Mach number $0<M_0\leq 1$ and total compressor pressure ratio $1\leq\pi_c\leq 40$ are considered as design variables [8].

.

III. Multi-objective optimization

A general minimization problem of M objectives can be mathematically stated as: given: $\vec{x}=[x_1,x_2,...,x_d]$ Where d is the dimensional of the decision variable space, Minimize:

$$\vec{f}(\vec{x})=f_i(\vec{x}) \qquad i=1,...,M \tag{1}$$

According to J number condition:

$$g_j(\vec{x})\leq 0 \qquad j=1,...,J \tag{2}$$

And K number of equal condition:

$$h_k(\vec{x})=0 \qquad k=1,...,K \tag{3}$$

The MOO problem then reduces to finding an $\vec{x}$ such that $\vec{f}(\vec{x})$ is optimized. Since the notion of an optimum solution in MOO is different compared to the SOO, the concept of Pareto dominance is used for the evaluation of the solutions. This concept formulated by Vilfredo Pareto is defined as [7]:

A vector $\vec{u}=(u_1,u_2,...,u_M)$ is said to dominate a vector $\vec{v}=(v_1,v_2,...,v_M)$, (denoted by $\vec{u}\leq\vec{v}$), for a multi objective minimization problem, if and only if

$$\forall i\in\{i,...,M\},\quad u_i\leq v_i \wedge \exists i\in\{1,...,M\}\ :u_i<v_i \tag{4}$$

where M is the dimension of the objective space.

A solution $\vec{u}\in U$, where U is the universe, is said to be Pareto Optimal if and only if there exists no other solution $\vec{v}\in U$, such that $\vec{u}$ is dominated by $\vec{v}$. Such solutions $(\vec{u})$ are called non-dominated solutions. The set of all such non-dominated solutions constitutes the Pareto-Optimal Set or non-dominated set.

A. *Multi-Objective Particle Swarm Optimization (MOPSO) Algorithm*

The PSO algorithm is an optimization technique based on a bird migration pattern. In the real world, the movement of birds towards food can display a regular system in which each bird improves its position in the time dimension. In the PSO algorithm, each bird can be presented as a particle (single solution) and a set of the birds is identified as a swarm (set of solutions). Thus, in an optimization problem, the position of the *i*th particle (x_i) can be represented by a D-dimensional vector:

$$x_i=[x_{i1},x_{i2},...,x_{iD}] \qquad i=1,...N \tag{5}$$

where D= number of decision variables and N= swarm size. Moreover, the best bird (with the smallest distance from the food) is called the global best (Gbest), and the best position of a bird ever tried toward the food is the particle best (Pbest). Steps of the PSO algorithm are as follows:

In the first step, random solutions (x_i) with a normal distribution of decision variables are produced. In the second step, those solutions are used in the simulation model as input data. The objective function for each particle (solution) is then calculated and stored in the memory of the algorithm. For the next step, Pbest and Gbest are assigned with respect to the best position of particles and swarm so far discovered, respectively. In the fourth step, the velocity is calculated as:

$$v_{id}^{t+1}=\alpha(w^t.v_{id}^t+c_1.r_1^t(Pbest_{id}^t-x_{id}^t)+c_2r_2^t(Gbest_d^t-x_{id}^t))$$
$$i=1,2,...N,\ d=1,2,...D \tag{6}$$

$$w^t=w_{Max}-\frac{(w_{Max}-w_{Min})}{Iter_{Max}}\times t \tag{7}$$

In which: v_{id}^{t+1} = velocity of *i*th particle for *d*th dimension in *(t + 1)*th iteration.

α = constriction factor which is a fixed pre-specified value and controls the velocity of particles in the decision variables space.

w_t = inertia weight parameter in *t*th iteration which is calculated by Eq. (7).

This parameter starts from the maximum value (w_{Max}) in the first iteration to the minimum value in the last iteration $(iter_{Max})$. That is, at the beginning of search process, the effect of velocity (v_{id}^t) is more than that in later iterations. c_1 =cognitive parameter; c_2 =social parameter (c_1 and c_2 assign the proportion of Pbest and Gbest in the velocity); r_1^t and r_2^t = uniformly distributed random numbers in [0, 1] in the *t*th iteration. Hence, each particle moves in the decision space by a velocity vector with two elements. Thus, $Pbest_{id}^t$ = best position of the ith particle for the *d*th dimension in the *t*th iteration and $Gbest_d^t$ = best position of the swarm for the *d*th dimension in the *t*th iteration.

At the next step, resulting velocities are controlled using lower $(w_{\min})$ and upper $(w_{\max})$ bounds of velocity:

$$v_{\min}\leq v_{id}^{t+1}\leq v_{\max} \tag{8}$$

Finally, the particle position is calculated by using Eq. (9):

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \quad (9)$$

The resulting particle position is used as the new input for the simulation model in the second step and new objective functions are again calculated. This process continues up to the maximum number of iterations. The aforementioned PSO algorithm has a continuous search mechanism to move toward an optimal point. The movement toward an optimal point in the single-objective PSO (SOPSO) algorithm is different from that in the MOPSO algorithm. In the SOPSO algorithm, each particle follows its objective function in the search process. The vector evaluated PSO (VEPSO) is a multi-objective algorithm that uses one swarm for each objective. Thus, each swarm uses its particle position as the Pbest, but the Gbest of each swarm is replaced by the Gbest of other swarms for the next iteration.

B. Defining Pareto Front

Vectors including target functions which are made from vectors of Pareto collections (Θ*) are called Pareto Front (Ω is an accepted design region which satisfy Eq. (2) and Eq. (3).).

$$\Theta^* = \{X \in \Omega \mid \exists X' \in \Omega : F(X') \prec F(X)\} \quad (10)$$

The results of multi target optimization have no superiority toward each other and are called non superior results. In Fig. 1, for example can see the Pareto points, in this figure by moving from A to B (or vise versus), any improvement in condition of any target functions can deteriorate the condition of at least one target function of problem, (the goal is to minimize or maximize both target functions). Pareto optimum points almost are located in boundary lines of design region or are over lapped points of target functions. In Fig. 1 the bold line shows such boundary line of two target functions which its component points are called Pareto Front.

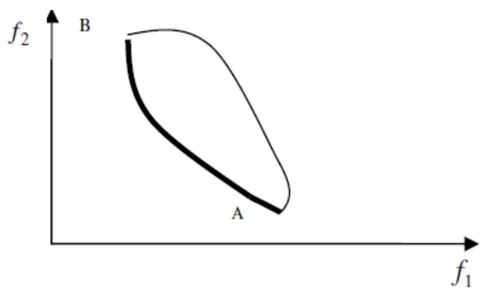

Figure 1. Pareto points in a curve form

IV. OPTIMIZATION ACCORDING TO MOPSO

A. Optimization according to special thrust and special fuel consumption target functions (ST, SFC)

Pareto point's collections in Fig. 2, is shown according to target functions and by using MOPSO algorithm.

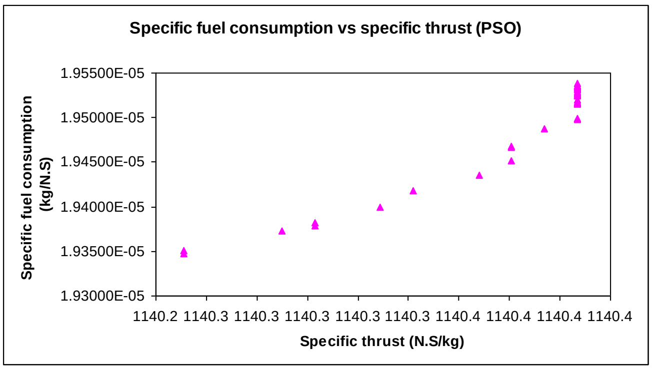

Figure 2. Pareto points of specific thrust and specific fuel consumption

Comparing one and two design vectors, assist us to conclude that by increase in 0.00014% of the special thrust, fuel consumption will be escalated up to 0.01%. By considering Pareto curves, we observe that point expansion is in the limited space (has a small range). It means that MOPSO algorithm tends to being close to the best point of optimization. Fig.3 and Fig.4 show the Pareto points collection according to compressor pressure ratio (π_c) and flight Mach number (M_0).

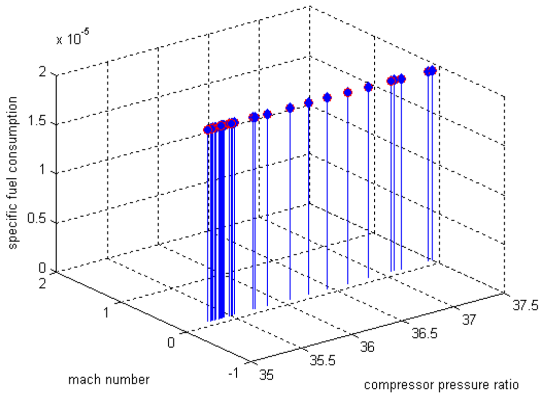

Figure 3. Pareto points of specific fuel consumption vs Mach number and compressor pressure ratio according to specific thrust and specific fuel consumption optimization

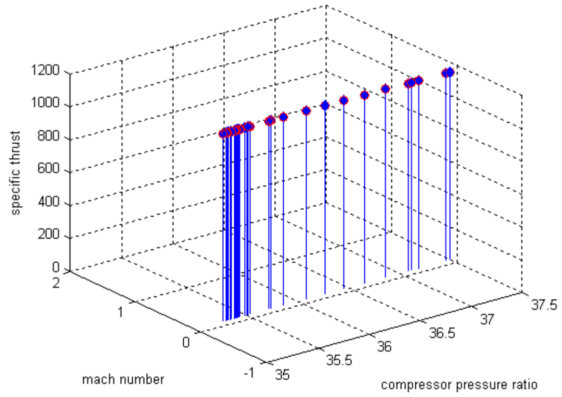

Figure 4. Pareto points of specific thrust vs Mach number and compressor pressure ratio according to specific thrust and specific fuel consumption optimization

B. Optimization according to special thrust and thermal efficiency target functions (ST, η_t)

Pareto points of these target functions are shown in Fig. 5, by using MOPSO method. Comparing one and two design vectors reveals that by increasing 0.28% of special thrust, thermal efficiency will be decreased up to 0.093%.

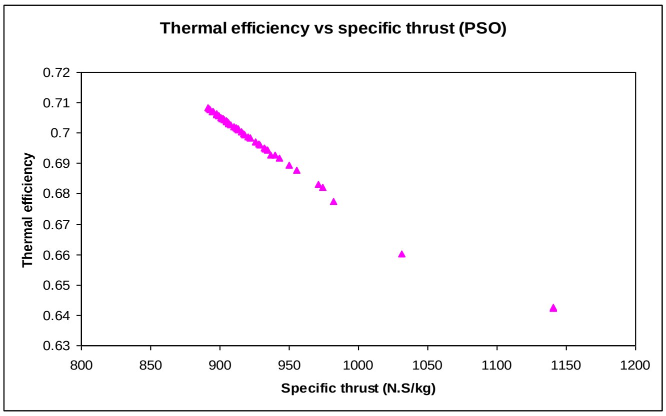

Figure 5. Pareto points of specific thrust and thermal efficiency

Since the goal is increasing the thrust and thermal efficiency and also by considering the fact that decreasing thermal efficiency compare to increasing thrust is negligible, therefore the second design vector should be considered more valuable than the first one.

Pareto points' sets of these target functions are shown in Fig. 6 and Fig. 7, based on compressor pressure ratio and Mach number.

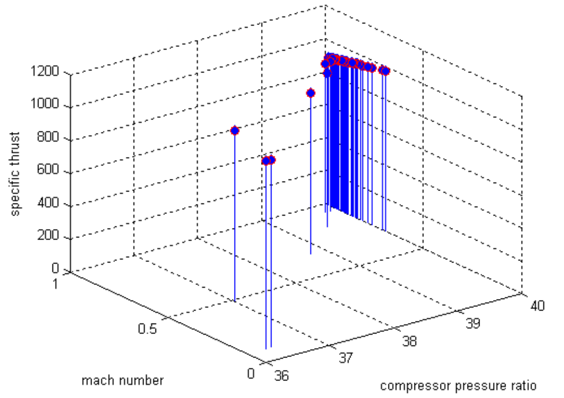

Figure 6. Pareto points of specific thrust vs Mach number and compressor pressure ratio according to thrust and thermal efficiency optimization

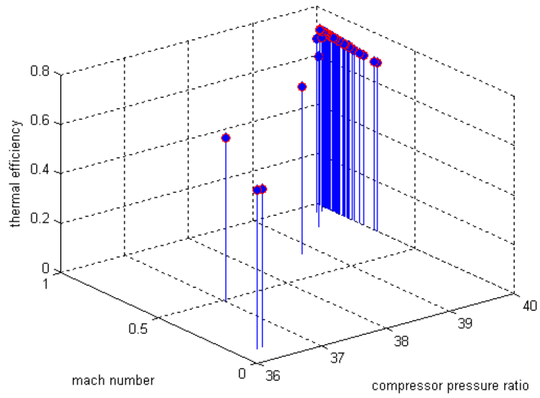

Figure 7. Pareto points of thermal efficiency vs Mach number and compressor pressure ratio according to thrust and thermal efficiency optimization

C. Optimization according to special fuel consumption and thermal efficiency target functions (SFC, η_t)

Fig. 8 shows the Pareto curve according to these target functions. Comparing one and two design vectors help us to reach to the conclusion that by 0.00055% growth in specific fuel consumption, thermal efficiency will be increased up to 0.0011%. By considering Pareto curves, we observe that point expansion has a small range. It means that MOPSO algorithm tends to being close to the best point of optimization.

Pareto points' sets of these target functions are shown in Fig. 9 and Fig. 10 based on compressor pressure ratio and Mach number.

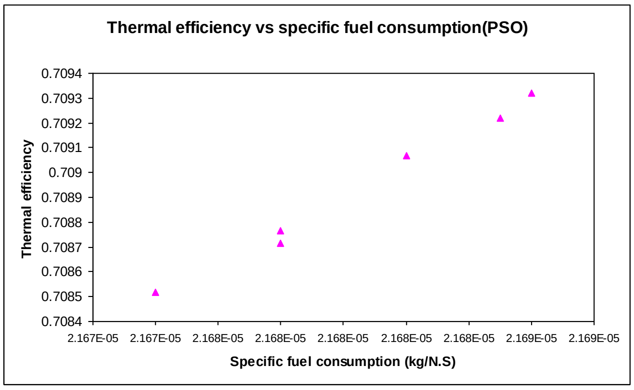

Figure 8. Pareto points of specific fuel consumption and thermal efficiency

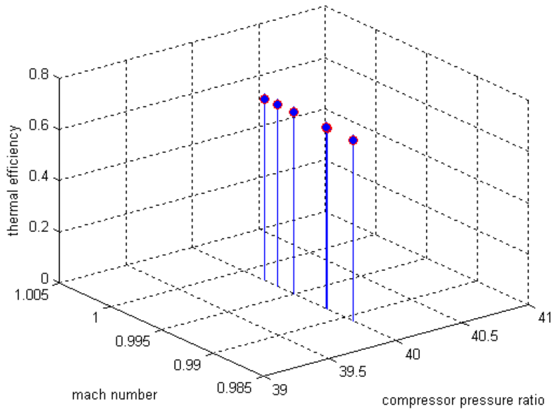

Figure 9. Pareto points of thermal efficiency vs Mach number and compressor pressure ratio according to thermal efficiency and specific fuel consumption optimization

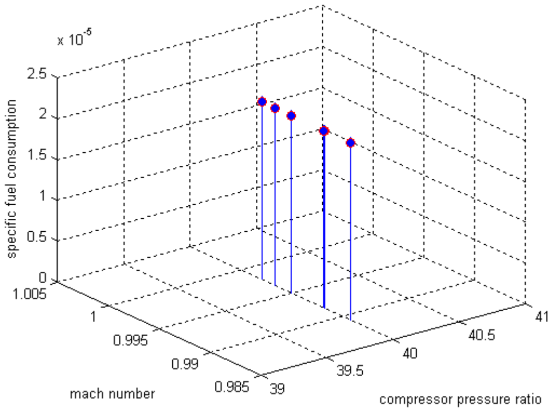

Figure 10. Pareto points of specific fuel consumption vs Mach number and compressor pressure ratio according to thermal efficiency and specific fuel consumption optimization

V. CONCLUSION

Below, the results of MOPSO optimization method are collected.

It can be seen that specific thrust is more affected by compressor pressure ratio; it means that by neglecting Mach number, wherever the compressor pressure ratio is high, the specific thrust is increasing.

By considering the Pareto curves more carefully, we reach to the point that specific fuel consumption has almost a direct relationship with square of specific thrust $(SFC \alpha ST^2)$.

Also by notifying Pareto points of thermal efficiency and specific thrust, we conclude that the relationship between these two target functions is a linier relationship or on the other hand $\eta_t \alpha\ m(ST)$ where m is $(-1 < m < 0)$.

Based on designed Pareto points we can see that Pareto points expansion of MOPSO optimization method is small. This characteristic can be considered as a power of MOPSO which these Pareto points is tended to keep close to the best answer.

Thermal efficiency (η_t), Specific thrust (ST)

$$0.64 < \eta_t < 0.71$$
$$891.58 \le ST \le 1140.37$$
$$0.1 \le M_0 \le 0.98$$
$$36.3 \le \pi_c \le 40$$

Specific fuel consumption (SFC), Specific thrust (ST)

$$1.93 \le SFC \times 10^5 < 1.95$$
$$1140.25 < ST < 1140.41$$
$$M_0 = 0.1$$
$$35 < \pi_c < 37.5$$

Specific fuel consumption (SFC), Thermal efficiency (η_t)

$$0.7 < \eta_t < 0.71$$
$$2.1 < SFC \times 10^5 < 2.2$$
$$0.98 \le M_0 < 1$$
$$\pi_c = 40$$

APPENDIX I

Symbol	Quantity	unit
a_0	Velocity of sound at inlet	m/s
$\dot{m}_0$	Mass flow rate	Kg/s
T_{t4}	Burner exit total temperature	K
h_{pr}	Heating value	kJ/kg
V_0	Air velocity at inlet	m/s
g_c	Newton's constants	kg-m.(N-s^2)$^{-1}$
R	Gas constants	J/kg.K
F	Thrust	N
M_0	Flight Mach number	
τ_r	Total static temperature ratio at inlet	
τ_t	Burner exit/inlet total temperature ratio	
τ_λ	Burner exit total enthalpy/inlet total enthalpy	
τ_c	Compressor exit total temperature/Compressor inlet temperature	
f	Fuel/air ratio	
ST	Specific thrust	
SFC	Specific fuel consumption	
η_t	Thermal efficiency	
π_c	Total compressor pressure ratio	

REFERENCES

[1] C.A.C. Coello, M.S. Lechuga, MOPSO: A proposal for multiple objective particle swarm optimization, in: Proceedings of the 2002 Congress on Evolutionary Computation part of the 2002 IEEE World Congress on Computational Intelligence, IEEE Press, Hawaii, 2002, pp. 1051–1056, May.

[2] C.A.C. Coello, G.T. Pulido, M.S. Lechuga, Handling multiple objectives with particle swarm optimization, IEEE Transactions on Evolutionary Computation 8 (3) (2004) 256–279.

[3] D.W. Corne, N.R. Jerram, J.D. Knowles, M.J. Oates, PESA-II: region-based selection in evolutionary multi objective optimization, in: Proceedings of the Genetic and Evolutionary Computing Conference (GECCO-2001), Morgan Kaufman, 2001, pp. 283–290.

[4] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II, IEEE Transactions On Evolutionary Computation 6 (2) (2002) 182–197.

[5] A.P. Engelbrecht, Fundamentals of Computational Swarm Intelligence, John Wiley and Sons, USA, 2006, January.

[6] J.E. Fieldsend, S.Singh, A multi-objective algorithm based upon particle swarm optimization, an efficient data structure and turbulence, in: Proceedings of UK Workshop on Computational Intelligence (UKCI'02), vol. 2–4, Bermingham, UK, September 2002, pp. 37–44.

[7] X. Hu, R. Eberhart, Multiobjective Optimization Using Dynamic Neighbourhood Particle Swarm Optimization, in: Proceedings of the 2002 Congress on Evolutionary Computation part of the 2002 IEEE World Congress on Computational Intelligence, IEEE Press, Hawaii, 2002, May.

[8] Jack D. Mattingly, 1996. Elements of Gas Turbine Propulsion, McG-raw-Hill series in mechanical engineering.

[9] J. Kennedy, R. Eberhart, Particle swarm optimization, in: IEEE International Conference Neural Networks, 1995, pp. 1942–1948.

[10] X. Li, A non-dominated sorting particle swarm optimizer for multi-objective optimization, in: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2003), Lecture Notes in Computer Science, vol. 2723, Springer, 2003, pp. 37–48.

[11] S. Mostaghim, J. Teich, Strategies for Finding Good Local Guides in Multi-objective Particle Swarm Optimization (SIS'03), IEEE Service Center, Inidanapolis, Indiana, USA, 2003. April 26–33.

[12] K.E. Parsopoulos, M.N. Vrahatis, Particle swarm optimization method in multiobjective problems, in: Proceedings of the 2002 ACM Symposium on Applied Computing (SAC 2002), 2002, pp. 603–607.

[13] J.D. Schaffer, Some experiments in machine learning using vector evaluated genetic algorithm, Ph.D. thesis, Vanderbilt University, Nashville,TN, 1984.

[14] F. van den Bergh, A.P. Engelbrecht, A study of particle swarm optimization particle trajectories, Information Sciences 176 (2006) 937–971.

[15] Y. Zhang, S. Huang. A novel multi-objective particle swarm optimization for buoys-arrangement design, in: Proceedings of the IEEE/ WIC/ACM International Conference on Intelligent Agent Technology (IAT 2004), 2004, pp. 24–30.

[16] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm. technical report TIK-103, Computer Engineering and Network Laboratory (TIK), Swiss Fedral Institute of Technology (ETH), Gloriastrasse 35, CH-8092 Zurich, Swidzerland, May 2001.

APPENDIX II [8]

$$R = \frac{\gamma - 1}{\gamma} C_P$$

$$a_0 = \sqrt{\gamma R g_c T_0}$$

$$\tau_r = 1 + \frac{\gamma - 1}{2} M_0^2$$

$$\tau_\lambda = \frac{T_{t4}}{T_0}$$

$$\tau_c = (\pi_c)^{(\gamma-1)/\gamma}$$

$$\tau_t = 1 - \frac{\tau_r}{\tau_\lambda}(\tau_c - 1)$$

$$\frac{V_9}{a_0} = \sqrt{\frac{2}{\gamma - 1}\frac{\tau_\lambda}{\tau_r \tau_c}(\tau_r \tau_c \tau_t - 1)}$$

$$ST = \frac{F}{\dot{m}_0} = \frac{a_0}{g_c}\left(\frac{V_9}{a_0} - M_0\right)$$

$$f = \frac{C_P T_0}{h_{pr}}(\tau_\lambda - \tau_r \tau_c)$$

$$SFC = \frac{f}{ST}$$

$$\eta_t = 1 - \frac{1}{\tau_r \tau_c}$$