International Journal of Hybrid Information Technology
Vol.6, No.5 (2013), pp.269-276
http://dx.doi.org/10.14257/ijhit.2013.6.5.25

Multi-Objective Particle Swarm Optimization of Regenerative Intercooled Gas Turbine Cycle

Karim Salahshoor[1] and Kamal Jafarian[2]

[1]Electrical Engineering Department, Petroleum University, Tehran, Iran
[2]M.S. Student of Automation and Instrumentation Eng'g, Petroleum University, Ahwaz, Iran

salahshoor@put.ac.ir, Kamaljafarian67@gmail.com

Abstract

Gas turbines can be found in many industrial application areas. Gas turbine generation is limited by some undesirable effects which can be incorporated as operational constraints. Because of importance of energy, optimization of power generation systems is necessary. In order to achieving higher efficiencies, some propositions are preferred such as recovery of heat from exhaust gases in a regenerator, utilization of intercooler in a multistage compressor, steam injection to combustion chamber and etc. In this article multi-objective particle swarm optimization are employed for Pareto approach optimization of Gas Turbine cycle. In the multi-objective optimization a number of conflicting objective functions are to be optimized simultaneously. Multi-objective optimization offers a candidate scheme whose solution can satisfy the foregoing major requirements. At the first stage single objective optimization has been investigated and then MOPSO has been used for multi-objective optimization. The sets of selected decision variables based on this Pareto front, will cause the best possible combination of corresponding objective functions. The obtained results show that the output of multi-objective optimization scheme confirms that of single objective results.

Keywords: *Gas turbine, Multi-objective optimization, Particle swarm optimization, Power generation*

1. Introduction

Most optimization problems in everyday life are not static in nature, have multiple objectives and at least two of the objectives are in conflict with one another. Multi-objective optimization problems (MOOPs) with conflicting objectives do not have a single solution. Therefore, MOO algorithms aim to obtain a diverse set of non-dominated solutions, i.e. solutions that balance the trade-off between the various objectives, referred to as the Pareto-optimal front (POF). Another goal of multi-objective algorithms (MOAs) is to find a POF that is as close as possible to the true POF of the problem. Many MOAs store the found non-dominated solutions in an archive. Therefore, if an algorithm finds new non-dominated solutions, the new solutions are compared with the solutions in the archive. If a new solution is dominated by any of the solutions in the archive, it is not placed in the archive. Otherwise, the new solution is placed in the archive and any solutions in the archive that are dominated by the new solution are removed from the archive/repository.

There are many methods to solve multi-objective problems. In this paper we use the multi-objective particle swarm optimization (MOPSO) algorithm. MOPSO is a PSO based

ISSN: 1738-9968 IJHIT

algorithm adapted for MOO by Lechuga [1]. In this paper, an optimal set of design variables in a gas turbine power plant, namely, compressor pressure ratio (Rp), excess air in combustion (EA), turbine inlet temperature (TIT), and inlet air temperature(T0) are used to reach Pareto front. Our considerable objective functions are net output power of cycle (W_{net}) and cycle thermal efficiency (η_T). Our aim is to optimize this objective functions, with regarding suitable practical constraints, using MOPSO.

From this section, input the body of your manuscript according to the constitution that you had. For detailed information for authors, please refer to [1].

2. Multi-Objective Optimization

2.1 Definitions

Each optimization problem contains one or more objective functions and a set of decision variables and most optimization problems contain a set of constraints. Optimization problems can be classified according to a number of characteristics, including the number of decision variables, the type of decision variables, the degree of linearity of the objective functions, the type of constraints, the number of optimization criteria or objectives and the number of optima [2, 3]. These concepts are discussed in more detail below.

The **objective function** represents the quantity to be optimized, *i.e.*, the quantity to be minimized or maximized. The objective function is also referred to as the *cost function* or *optimization criterion*. If the problem that has to be optimized is expressed using only one objective function, it is referred to as a SOOP. However, if a problem has more than one objective that has to be optimized simultaneously, it is called a MOOP. Each objective function has a vector of **decision variables** that influence the value of the objective function. Therefore, a search algorithm iteratively modifies the value of these variables to find the optimum for the objective function. If an optimization problem has constraints, the set of **constraints** restricts the values that can be assigned to the set of decision variables. When solving an optimization problem with equality or inequality constraints, the optimization method’s goal is to assign values from the specified domain to the decision variables in order to optimize the objective function and to satisfy the constraints.

Multi-objective optimization problems have more than one objective. Let a single objective function be defined as $f_k : R^{nx} \rightarrow R$. Then $f(x) = (f_1(x), f_2(x), . . . , f_{nk}(x)) \in O_{space} \subseteq R^{nm}$ represents an objective vector containing n_k objective function evaluations, and O_{space} is the *objective space*. Using the notation defined above, a MOOP can be mathematically defined as follows:

$$\begin{aligned}&\text{Minimize : } f(x)\\&\text{Subject to : } g_i(x) \leq 0,\ i = 1, . . . , n_g\\&h_j(x) = 0,\ j = 1, . . . , n_h\\&x \in [x_{min}, x_{max}]^{nx}\end{aligned} \quad (1)$$

For SOOPs, where only one objective is optimized, global optima are defined as the best candidate solutions that lead to the smallest value of the objective function. However, when dealing with a MOOP, the various objectives are normally in conflict with one another, *i.e.*, improvement in one objective leads to a worse solution for at least one other objective. MOOPs do not have specific optima, but trade-off solutions.

A decision vector x^* is Pareto-optimal if there does not exist a decision vector $x \neq x^* \in F$ that dominates x^*, *i.e.*, $\nexists k : f_k(x) < f_k(x^*)$. If x^* is Pareto-optimal, the objective vector, $f(x^*)$, is also Pareto-optimal.

Together, all the Pareto-optimal decision vectors form the Pareto-optimal set (POS), defined as the set of all Pareto-optimal decision vectors, *i.e.*,

$$P^* = \{x^* \in F \mid \nexists x \in F : x \prec x^*\} \qquad (2)$$

The POS contains the best trade-off solutions for the MOOP. The corresponding objective vectors form the Pareto-optimal front (POF), which is the objective vector f (x) and the POS P^*, the POF, $PF^* \subseteq O_{space}$ is defined as

$$PF^* = \{f = (f_1(x^*), f_2(x^*), \ldots, f_{nk}(x^*))\}, \forall x^* \in P^* \qquad (3)$$

Therefore, the POF contains the set of objective vectors that corresponds to the POS, i.e. the set of decision vectors that are non-dominated.

When solving a MOOP, the goal is to approximate the true POF. If the problem requires a single solution, the best trade-off solution is selected for the specific problem from the set of solutions represented by the POF. Therefore, the goal is to find an approximation of the true POF such that:

• The distance between the found POF and the true POF is minimized.

• The set of non-dominated solutions is as diverse as possible and as evenly spread out along the found POF as possible.

• The set of non-dominated solutions contains as many solutions as possible.

• The solutions that have been found and that form the found POF are stored for later reference.

2.2 Particle Swarm Optimization

Inspired by the social behavior of bird flocks, Eberhart and Kennedy [4] introduced PSO. The PSO algorithm (see Table 1) maintains a swarm of particles, where each particle represents a solution of the optimization problem under consideration. Each particle moves through the search space and the particle's position in the search space is updated based on its own experience (cognitive information), as well as the experience of its neighbors (social information). The particle's position that produced the best solution so far is referred to as its personal best or **pbest**. The position that lead to the best overall solution by all particles in a pre-defined neighborhood, *i.e.*, either the best of the neighborhood's particles' *pbests* or the best of the current positions of the neighborhood's particles, is called the neighborhood best or **nbest**. The first PSOs introduced by Eberhart and Kennedy are the global best PSO, or **gbest** PSO, and the local best PSO, or **lbest** PSO. The *gbest* PSO defines the neighborhood of each particle as the whole swarm. In this case the neighborhood best is also referred to as the global best or *gbest*.

The velocity of a particle is calculated as follows:

$$v_i(t+1) = v_i(t) + c_1 r_1(t)[y_i(t) - x_i(t)] + c_2 r_2(t)[y_N(t) - x_i(t)]$$

where $v_i(t)$ and $x_i(t)$ are the velocity and position of particle i at time step t respectively; $y_N(t)$ represents the *nbest* of neighborhood and $y_i(t)$ represents the *pbest* at time t; $c_1r_1(t)[y(t)-x(t)]$ is the cognitive component of the velocity and $c_2r_2(t)[y_N(t)-x(t)]$ is the social component of the velocity; c_1 and c_2 are positive acceleration coefficients that influence the contributions of the cognitive and social components respectively; and r_1, r_2 ~ $U(0, 1)^{nx}$ are random values sampled from an uniform distribution with n_x representing the number of decision variables or the dimension of the search space. Once the new velocity of a particle has been calculated, its new position can be determined by adding the velocity to its current position as follows:

$$x_i(t + 1) = x_i + v_i(t + 1) \qquad (4)$$

Table 1. Particle swarm optimization algorithm

Steps	
1	Create and initialize a swarm
2	While stopping condition has not been reached
3	For each particle in swarm do
4	Set *pbest*
5	Set *nbest*
6	For each particle in swarm do
7	Calculate new velocity
8	Calculate new position

2.3 Multi-Objective Particle Swarm Optimization

The MOPSO algorithm was introduced by Coello Coello and Salazar Lechuga [5] as one of the first PSO algorithms extended for MOO. Before the MOPSO algorithm can be executed, the swarm is initialized. Similar to PSO, the first step of the MOPSO algorithm initializes each particle's initial position, velocity and *pbest*, and sets swarm size, neighborhood size, and the control parameters.

In addition to the PSO initialization, the particles are evaluated and the positions of the particles that are non-dominated are stored in the archive. Furthermore, the search space that has been explored so far is divided into hypercubes and all particles are placed in a hypercube based on the particle's position in objective space.

MOPSO uses an archive to preserve elitism. However, the original version of MOPSO struggled to converge to the true POF in the presence of many local POFs [6]. To overcome this problem, Coello *et al.*, [6] introduced an updated version of MOPSO that uses a mutation operator. Initially, the mutation operator is applied to all particles, but then the number of particles being mutated decreases rapidly as the number of iterations increases. The goal of the mutation operator is to increase the swarm's exploration ability. However, the mutation operator is not only applied to the particles, but also to the range of each decision variable of the MOOP. This leads to the whole range of each decision variable to be included in the beginning of the search, but then as the number of iterations increases, the range of each decision variable decreases. Coello *et al.*, [6] compared the performance of the MOPSO with the mutation operator against three other MOO algorithms, namely NSGA-II, Micro-GA [7] and pareto archived evolution strategy (PAES) [8] on five constrained MOO benchmark functions. The results of the study indicated that MOPSO with the mutation operator was the only MOO algorithm able to find solutions along the full extent of the POF for all benchmark functions (see Table 2).

Table 2. Multi-objective PSO algorithm

Steps	
1	Create and initialize a swarm
2	While stopping condition has not been reached
3	Calculate new velocity
4	Calculate new position
5	Manage boundary constraint violations
6	Update archive
7	Update the particles' allocation to hypercubes
8	For each particle in swarm do
9	Update *pbest*

3. The Gas Turbine Power Plant

A schematic of RIGT (Regenerative-Intercooled-Gas Turbine) cycle is given in Figure 1. The system consists of a two-stage intercooled air compressor, a regenerator, a combustion chamber and a gas turbine. The incoming air has a pressure of 1.013 bars. Turbine and compressor have an isentropic efficiency of 87 and 85 percent, respectively. The regenerative heat exchanger has an effectiveness of 75%. Combustion chamber adiabatic efficiency is 98%.the pressure drop through the air preheater is 4% of the inlet pressure for both flow streams and through the combustion chamber is 3% of the inlet pressure. It is 2% for intercooler. The fuel (natural gas, type C, $C_{1.5}H_5$), is injected at environment temperature and pressure slightly more than environment pressure. In our cycle, overall compressor pressure ratio is R_p = P4/P1 and for each stage, pressure ratios are RP_1 and RP_2, respectively. Temperature of hot air exiting from first stage reduces to compressor inlet air temperature, due to heat extraction in intercooler.

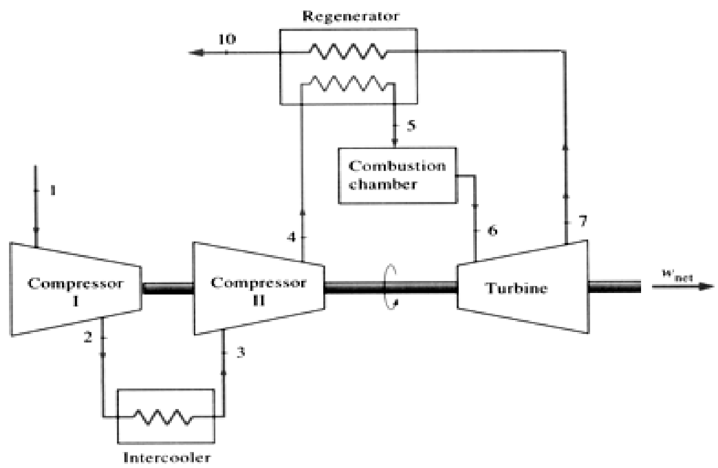

Figure 1. Regenerative-Intercooled Gas Turbine Cycle

4. Optimization Problem

Our goal, as mentioned before, is to maximize $\eta(R_p,TIT,EA,T_0)$ and $W_{net}(R_p,TIT,EA,T_0)$ simultaneously. In our study, according to selection of R_p, TIT, EA, T_0 as design parameters, suitable practical constraints must exert on objective function. These constraints are selected regarding to responsible references and sources. Our linear constraints are:

- Compression ratios between 3 to 15 are used at modern gas turbine cycles. Higher amount of this parameter is used for propulsive gas turbine cycles. In common power stations, compression ratio is bounded between 11 to 16. So, for considering wide range of compression ratios, we select it as follow: $1 \leq R_p \leq 25$.

- Constraint of maximum temperature of cycle, is metallurgical nor thermodynamically. Presently maximum turbine inlet temperature is about 1250 to 1340^{o}C. In modern gas turbine cycles, this temperature is about 1500^{o}C .Therefore: $1200 \leq TIT \leq 1600$ K.

Table 3. Values of decision variables in SOOP using PSO

W_{net} = 721.2145	η_T = 48.71 %
R_P = 24.8219	R_P = 23.3721
T_0 = 261.5125	T_0 = 262.7014
TIT = 1567.3	TIT = 1589.4
EA = 1.0028	EA = 1.0029

This objective function is limited with two nonlinear constraints, too. First, in order to ease in natural displacement of exhaust gases (produced in combustion process) in stacks due to inequality of densities, and ecological considerations, we have [9]: $Exh \geq T_0$. Also, because of presence of some compositions such as Nitrogen, Sulfur and etc. in combustion productions, and for avoiding formation of corrosive materials such as sulfuric acid, nitric acid, *etc.*, we must restrain formation of water drops in cycle exhaust. For this purpose we consider this constraint as [10]: Exh ≥ $T_{dewpoint}$ which $T_{dewpoint}$ is the dew point of combustion products. Finally, according to these constraints, desired nonlinear objective function is optimized using genetic algorithm in MATLAB. The results of the single-objective optimization using PSO are summarized in Table 3.

Figure 2 shows Pareto fronts of two objectives after using MOPSO scheme. As you see interval variation is (45.29, 48.71) and (703.1, 721.2) for thermal efficiency and net power output, respectively. Two end points of Pareto optimal solution are the same optimum values in each direction. So MOPSO scheme's output confirms single-objective optimization results appropriately.

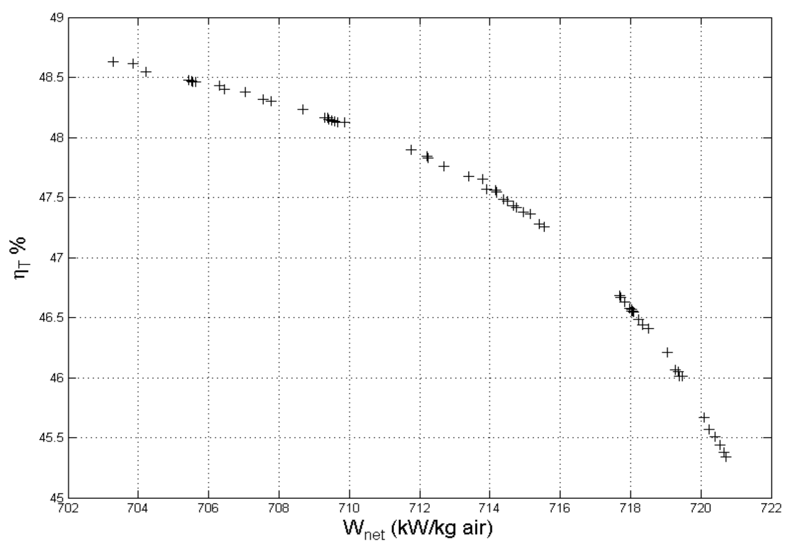

Figure 2. Pareto front of thermal efficiency and net power

5. Conclusion

The work presented in this paper provided a provided a multi objective PSO (MOPSO algorithm) to obtain Pareto based optimization of the performance of a Brayton Cycle. Applying the multi-objective functions, namely thermal efficiency and the net output power were determined in terms of four design variables (Compressor pressure ratio, Excess air in combustion, turbine inlet temperature and inlet air temperature). Multi objective Particle Swarm Optimization MOPSO can easily handle constraints of gas turbine operation. It does not require a priori knowledge of the relative importance of the objective functions. Simultaneous optimization of two outputs revealed some interesting features among optimal objective functions and decision variables involved in the thermodynamic cycle of proposed system that would have not been obtained without the use of a multi-objective optimization approach. There is a set of acceptable trade-off near optimal solutions. This set is called Pareto front or optimality trade-off surfaces. It was also demonstrated that two extreme points in the Pareto included those of single objective optimization results.

References

[1] M. S. Lechuga, “Multi-objective optimization using sharing in swarm optimization algorithms”, PhD thesis, University of Birmingham, **(2009)** July.
[2] Y. Collette and P. Siarry, “Multiobjective optimization”, Decision Engineering Springer-Verlag, **(2003)**.
[3] A. P. Engelbrecht, “Fundamentals of Computational Swarm Intelligence”, John Wiley and Sons, Ltd., **(2005)**.
[4] J. Kennedy and R. C. Eberhart, “Particle swarm optimization”, In Proceedings of International Conference on Neural Networks, vol. IV, **(1995)**, pp. 1942–1948.
[5] C. A. C. Coello and M. S. Lechuga, “MOPSO: a proposal for multiple objective particle swarm optimization”, In Proceedings of Congress on Evolutionary Computation, vol. 2, **(2002)**, pp. 1051–1056.
[6] C. A. C. Coello, G. T. Pulido and M. S. Lechuga, “Handling multiple objectives with particle swarm optimization”, IEEE Transactions on Evolutionary Computation, vol. 8, no. 3, **(2004)** June, pp. 256–279.
[7] C. C. Coello and G. T. Pulido, “A micro-genetic algorithm for multiobjective optimization, Evolutionary multi-criterion optimization”, vol. 1993 of Lecture Notes in Computer Science, Springer Berlin/Heidelberg, **(2001)**, pp. 126–140.
[8] J. Knowles and D. Corne, “The Pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation”, In Proceedings of the Congress on Evolutionary Computation, vol. 1, (1999), p. 98–105.
[9] S. Farahat, E. K. Nejad, S. M. H. Sarvari, “Thermodynamic Optimization of Turboshaft Engine using Multi-Objective Genetic Algorithm”, World Academy of Science, Engineering and Technology, **(2009)**, pp. 782-788.
[10] M. Nikaein, "Thermodynamic optimization of Brayton cycle using exergy analysis with genetic algorithm", IEHT Journal, vol. 1, **(2009)**, pp. 63-68.

Authors

Karim Salahshoor

He received the M.S. and Ph.D. degrees in Electrical Engineering from University of Manchester Institute of Science and Technology in 1987 and 1990, respectively. During 1990-2013, he stayed in Petroleum University of Technology as a faculty member in Tehran, Iran. His research interest is process control and nonlinear system identification using artificial intelligence techniques.

Kamal Jafarian

He received the B.S. degree in mechanical engineering from Amirkabir University of Tech. in 2009. At now he is a M.S. student of automation and instrumentation engineering in Petroleum University of Tech. His research interest is optimal control theory in both classical and evolutionary forms.

