Multi-Objective Optimization of Solar Powered Irrigation System by Using Genetic Algorithm

by

Muhammad Ali Husaini bin Mohd Tholaat
16080

Dissertation submitted in partial fulfilment of
the requirements for the
Bachelor of Engineering (Hons)
(Chemical Engineering)

SEPTEMBER 2015

Universiti Teknologi PETRONAS,
32610 Bandar Seri Iskandar,
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Multi-Objective Optimization of Solar Powered Irrigation System by Using Genetic Algorithm

by

Muhammad Ali Husaini bin Mohd Tholaat
16080

A project dissertation submitted to the
Chemical Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(CHEMICAL ENGINEERING)

Approved by,

____
(Dr. Ir. Abdul Halim Shah bin Maulud)

UNIVERSITI TEKNOLOGI PETRONAS

BANDAR SERI ISKANDAR, PERAK

September 2015

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

MUHAMMAD ALI HUSAINI BIN MOHD THOLAAT

ABSTRACT

Irrigation system is synonym with agriculture. Conventional way of supplying source of energy to work the water pumping system is through fuel combustion such as diesel. Nowadays fuel combustion is not an attractive and feasible approach in a long run due to hiking fuel price and it is also not environmentally friendly which it may lead to pollution. The development of renewable energy such as solar energy as an external heat source rather be more attractive. However, this complex system needs to be optimized by using suitable metaheuristic technique in order to make the design to be economically and practically efficient. Thus, Genetic Algorithm is applied to solve multiple objective solar-irrigation system optimization. It is identified that the best setting should be input to get an optimal solution. Initial range of [1; 2] and crossover fraction of 1.0 have majorly contributed to the optimal search parameters. After some tuning to get the best setting, the simulation shows that the fitness function of 3 objectives resulted with 17.4303 kW power output, 15.2355% efficiency and $143,533.10 fiscal savings. This set of optimal solution is not as closed as other technique to the desired design objectives. Genetic Algorithm is a common technique and easy to work with but it has yet to be the best metaheuristic technique for this engineering problem due to some drawbacks.

ACKNOWLEDGEMENT

I am very grateful to Allah swt for the good health and wellbeing that were necessary for me to complete this project. I wish to express my sincere thanks to my project supervisor, Dr. Timothy Ganesan and Dr. Ir. Abdul Halim Shah bin Maulud, for providing me with all the necessary and endless guidance and brilliant advices for the project. I place on my best, sincerest gratitude to all my family members, for all the continuous, unceasing moral supports and encouragements given to me along the way from the beginning until the project completion.

I am also very thankful and indebted to all my fellow colleagues for sharing expertise, and sincere and valuable guidance and encouragement extended to me, especially in their extensive contributions in aiding me in dealing with the research tools and project writing. Last but not least, I would like to place on record, my sense of appreciation to one and all, who directly or indirectly, have lent their hand in this venture.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background of Study

Engineering problems are typically related to simultaneous multiple optimization of several goals and objectives. These objectives are often to be conflicting, inhibiting each objective to be simultaneously optimized (Konak, Coit, & Smith, 2006). The problems are complex and difficult, but may be solve by using the correct methods and techniques. In the agricultural industry, there are many types of irrigation system designed and invented all over the world. However, the systems implemented are encountering many problems as they become complex. For example, in adopting an automated water pumping system that works through combustion of fuels has made engineers to struggle to come out with combine automated devices which then works by itself with less human intervention. Furthermore, renewable sources of energy are nowadays more attractive to be embedded to the conventional way of cultivating crops. This makes this engineering problem becoming more complex and not easy to be optimized. Traditional way of supplying energy to operate the watering pump used to be power- grid motors and fuel-based engines. However, using fuel combustion to source the power is a major contribution to air pollution and carbon-based climate change. In addition, climbing fuel costs and energy self-sufficiency have made the development of feasible sources of clean alternative energy really crucial for most parts of the world (Kelley, Gilbertson, Sheikh, Eppinger, & Dubowsky, 2010).

The development of solar energy replacing fuel engines as an alternative sustainable energy is reasonably attractive. This system, however, needs to be technically and economically feasible. The feasibility may be dependent on many factors such as output power, overall system efficiency, cost savings etc. In a solar powered irrigation system, some of the design requires the conversion of heat energy to mechanical work through heat engine. Solar radiation is used to heat and evaporate the working fluid at high pressure, after which the vapour is expanded to generate mechanical shaft work. This mechanical work then will be used directly to drive a water pumping system. The versatility of the output and the potential to store solar heat (e.g. as hot water) presents a possible advantage over solar-photovoltaic for domestic heat and electricity load profile matching. Furthermore, the prospective to develop high-efficiency, low-cost components fit for the domestic scale could see upgraded competitiveness with photo-voltaic in the short-term. (Freeman, Hellgardt, & Markides, 2015)

Therefore, this multiple objectives need to be satisfied. Often, there is no single optimal solution, but rather a set of alternative solutions. These solutions are optimal in the wider sense that no other solutions in the search space are superior to them when all objectives are considered. They are known as Pareto-optimal solutions (Zitzler & Thiele, 1999). Genetic Algorithms are a popular meta-heuristic technique that is predominantly compatible for this class of problems.

1.2 Problem Statement

Water pumping by using diesel-based combustion used to be an attractive way out due to the high power range of the pumps. It may keep on pumping water to run into several demands over the day. However, the recent upswings in the fuel price and an elaborated and skilled care requirement of the diesel motor has made these systems to be an expensive solution for long term (Senol, 2012). Utilizing the renewable energy resources is proven to be an alternative way to solve the energy crisis and achieve the sustainable development of human beings due to their potentials in reducing fossil fuel consumption and improving environmental problems. Due to its non-polluting and wide-ranging prospects in applications has raised solar energy as a favourable clean renewable energy which attracted much attention particularly in recent years (Wang, Yan, Zhao, & Dai, 2014). The system also has decent ecological and economic performance in the agricultural site in comparison to the irrigation system driven by diesel engine (Gao et al., 2013).

The solar irrigation technology is to convert the solar radiation energy into electrical energy which raises water by driving the pump. Rankine cycle is operated by sourcing the heat energy from external source (i.e. solar energy captured by solar panel) to Rankine cycle evaporator or boiler. This system is an example of an engineering system that requires optimization of simultaneous objectives such as pump load/ power input, overall efficiency and fiscal savings (Chen, Tsui, Allen, & Mistree, 1994). Therefore, a suitable metaheuristic method of optimization needs to be implemented for this problem. Genetic algorithm is one of the common method in searching the optimal set of solution to the developed model formulation.

1.3 Objectives and Scope of Study

There are few objectives to be achieved in this research:

i. Identify and study multi-objective system for solar powered irrigation system.
ii. Identify suitable technique to solve multi-objective optimization of solar powered irrigation system.
iii. Implement Genetic Algorithms multi-objective optimization technique to solar powered irrigation problem and analyze results.

This paper will be analyzing the existing system of the solar powered irrigation system which incorporated the solar energy collection cycle with the heat engine cycle to convert the heat energy to the mechanical shaft work through Rankine cycle theory. Chen et. al (1994) has developed the solar irrigation problem and rigorously validated the model. They have derived the model formulation for this system and optimized by using DSIDES software during the year. The same system formulation was further taken into a different optimization technique by Ganesan et. al (2013). In that paper, they are using Analytical Programming approach with the aid of C++ language program. The same formulation taken for different optimization method which is Genetic Algorithm will be further analyzed and discussed in this paper.

CHAPTER 2

LITERATURE REVIEW

2.1 Irrigation system

Irrigation is always related with agriculture. In simple terminology, irrigation can be defined as the replacement or supplementation of rainwater with another source of water. It is a science of artificial application of water to the land or soil. The main idea behind irrigation systems is that the lawns and plants are maintained with the minimum amount of water required. Irrigation has been used for many purposes, among them are for maintenance of landscapes and revegetation of disturbed soils in dry areas and during periods of inadequate rainfall. However when relates to agriculture, irrigation is one of a major section to assist in the growing of agricultural crops.

Irrigation system has enter into sustainable development domain. Fuel combustion as the source of heat energy for producing shaft work is no longer attractive due to fuel price hikes and pollution. One of the typical design of irrigation system which embedded with solar energy as heat source are shown in figure below.

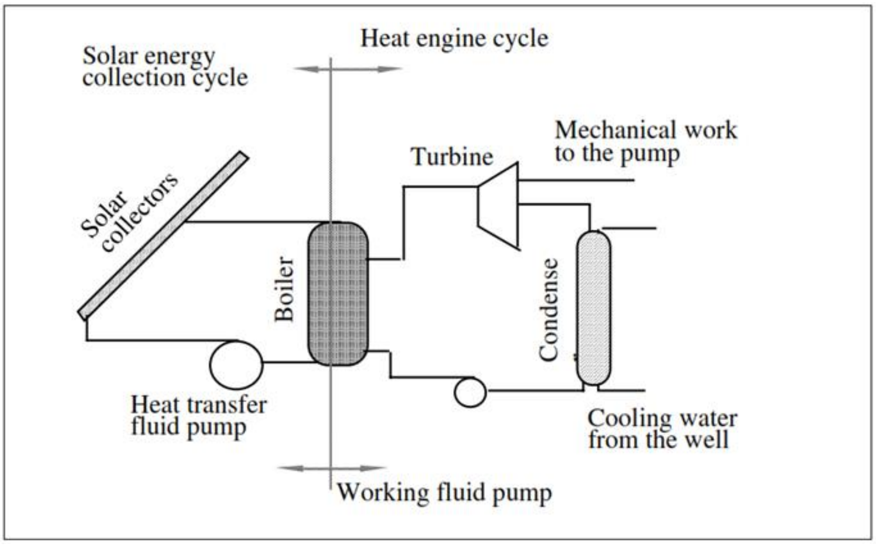

FIGURE 1: Solar Powered Irrigation System Model Configuration

This works with two energy cycle. The first part is the solar energy collection cycle. This section will supply heat energy externally to the other part of the system which is heat engine cycle driven by Rankine Cycle Theory. Heat transfer fluid pump will circulate the liquid from solar energy collection cycle to source the heat to the boiler or evaporator. The liquid water from the Heat Engine Cycle will be pumped to circulate the water to pass through the boiler. The saturated liquid water has increment in pressure and enters the boiler to be converted to saturated steam. The saturated steam drive the turbine to produce mechanical work which then supplied to the water pump for irrigation purposes. The saturated steam condensed and the process cycle repeated. For optimization purposes, 3 objectives to be maximized which are power output (should be approaching 20kW), efficiency (should be approaching 20%) and fiscal savings (maximized to $150,000).

2.2 Multi-Objective Optimization

In many real-life problems, objectives under consideration conflict with each other. Hence, optimizing a variable with respect to a single objective often results in unacceptable results with respect to the other objectives. Therefore, a perfect multi-objective solution that simultaneously optimizes each objective function is almost impossible. A reasonable solution to a multi-objective problem is to investigate a set of solutions, each of which satisfies the objectives at an acceptable level without being dominated by any other solution.

A solution is said to be Pareto optimal if it is not dominated by any other solution in the solution space. A Pareto optimal solution cannot be improved with respect to any objective without worsening at least one other objective. The set of all feasible non-dominated solutions in X is referred to as the Pareto optimal set, and for a given Pareto optimal set, the corresponding objective function values in the objective space is called the Pareto front. For many problems, the number of Pareto optimal solutions is enormous (maybe infinite).

The ultimate goal of a multi-objective optimization algorithm is to identify solutions in the Pareto optimal set. However, identifying the entire Pareto optimal set, for many multi-objective problems, is practically impossible due to its size. In addition, for many problems, especially for combinatorial optimization problems, proof of solution optimality is computationally infeasible. Therefore, a practical approach to multi-objective optimization is to investigate a set of solutions (the best-known Pareto set) that represent the Pareto optimal set as much as possible. With these concerns in mind, a multi-objective optimization approach should achieve the following three conflicting goals (Konak et al., 2006).

i. The best-known Pareto front should be as close possible as to the true Pareto front. Ideally, the best-known Pareto set should be a subset of the Pareto optimal set.
ii. Solutions in the best-known Pareto set should be uniformly distributed and diverse over of the Pareto front in order to provide the decision maker a true picture of trade-offs.
iii. In addition, the best-known Pareto front should capture the whole spectrum of the Pareto front. This requires investigating solutions at the extreme ends of the objective function space.

2.3 Multi-Objective Optimization in Engineering

There are many potential applications for genetic multi-objective optimization algorithms in engineering problems. For example, Belegundu, Murthy, Salagame, and Constants (1994) use them to design an airfoil and a laminated ceramic composite. The airfoil problem is based on the work of Kielb and Kaza in 1983, and it contains optimization of the torsional flutter margin to the maximum level as possible while at the same time minimizing the torsional resonant amplitude. The ratio of bending frequency to torsion frequency and the location of the center of gravity provide the two design variables, which are subject to limits. As for the ceramic composite lamination problem, the tensile stress in the core and the cost of material are minimized with stress constraints and some limitations on the design variables. Six design variables represent the volume fractions and thickness of different layers.

Garcia-Najera and Bullinaria (2009) have conducted multi-objective optimization for vehicle routing problem with time windows. It is a complex combinatorial optimization problem which can be understood as a fusion of two well-known sub-problems: the Travelling Salesman Problem and the Bin Packing Problem. Its main objective is to find the lowest-cost set of routes to deliver demand, using identical vehicles with limited capacity, to customers with fixed service time windows. The study has implemented a method to measure route similarity and incorporate it into an evolutionary algorithm.

Schaumann et al. (1998) has run an optimization on reinforced concrete structure and to an urban planning problem by using genetic algorithm. The construction time and material cost are minimized with the concrete structure. 112 design variables are used to represent the dimensions of 217 structural members. 98 additional variables are used to represent the number of workers needed to form the structural elements and to represent the delay in construction time. Constraints are imposed to limit the amount of steel reinforcement in each structural member. For the urban planning case, the optimization implicates the traffic travel time, cost and change in land used to be minimized. (Marler & Arora, 2004).

2.4 Genetic Algorithm Optimization

Genetic algorithm (GA) was first introduced in the 1960s by John Holland and developed by his students, friends and himself in the 1960s and the 1970s at the University of Michigan (Mitchell, 1995). GA has been the most popular heuristic approach to multi-objective design and optimization problems (Konak et al., 2006).

Genetic algorithm is a metaheuristic technique which is inspired the natural selection process, where stronger and fit individuals will survive in a competition. It also mimics the biological evolution. In nature each fellow of a certain population strives for water, food and territory, also struggle for attracting a mate is another aspect of nature. It is clear that the tougher individuals will deserve a better chance for reproduction and producing offspring, while the poor individuals will make less offspring or sometimes non. Consequently the gene of the strong or tough individuals will rise in the population. Offspring created by two fit individual (parents) has a potential to have a better fitness compared to both parents called super-fit offspring. By this norm the initial population changes to a better matched population to their environment in each generation (Amouzgar, 2012).

Two parents chosen by selection operator in the reproduction phase recombine to generate one or more children with mutation or crossover operators. They are few different crossover operators in literature but the main idea is picking two strings of solution (chromosomes) from the pool of selection operator and switching some portion from random selected points of these two strings. The application of mutation operator is done after cross over operator which genes are randomly changed to individual solutions in a string with a relatively small probability for a new chromosome to be generated. The purpose of this operator is to increase the likelihood of not dropping any potential solution, keep the diversity of the population and search for the global optimal, while cross over operator will rapidly explore the search space (Beasley, Martin, & Bull, 1993). In summary, the selection operator picks and sustains the good solutions; while crossover recombines the fit solutions to create a stronger and fitter offspring. Mutation operator on the other hand randomly modify a gene or genes in a string to optimistically search for a better string (Deb, 2001).

The genetic algorithm can be applied to solve problems that are not well suited for standard optimization algorithms, including problems in which the objective function is highly nonlinear, discontinuous, stochastic or non-differentiable. The genetic algorithm distinct from a classical optimization algorithm in two major ways:

TABLE 1: Comparison between classical and genetic algorithm

Classical Algorithm	**Genetic Algorithm**
Involve generation of a single point at each iteration. Optimal solution is approached by the sequence of points.	Involve generation of a population of points at each iteration. Optimal solution is approached by the best point in the population.
The next point in the particular sequence is selected by a deterministic computation.	The next population is selected by computation which uses random number generators.

CHAPTER 3

METHODOLOGY

In this chapter, the details of the methodological framework of this project are presented. The validity of the study of this project is judged and the steps as well as the procedures under taken on the way in fulfilment of the research objectives are presented. There were basically two important questions answered in this section which are how the data was collected or generated and how the generated data was analyzed. With regards to this section, Genetic Algorithm was the optimization toolbox that was used to generate solutions for the maximization of solar powered irrigation system. The results obtained from the GA were tabulated and analyzed in the results section. Therefore, the procedures employed to reach the solution to the problem are illustrated in this section.

3.1 Research Tools

The major tool that was used in this project is the MATLAB which is optimization and simulation software. MATLAB simply means matrix laboratory, a fourth generation of programming language and a multi-paradigm numerical computing environment. It was developed by Math work and allows matrix manipulation, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, such as C++, C, Java, FORTRAN and Python.

3.2 Genetic Algorithm Multi-objective Optimization Tool in MATLAB

Genetic algorithm is a stochastic optimization technique that searches for an optimal value of a complex objective function and are used to solve complicated optimization problems by simulation or mimicking a natural evolution process (Abimbola & Josiah, 2011). It involves repeated procedures with an initial population of potential solutions, a fitness evaluation via the application of genetic operators and the development of a new population. Abimbola and Josiah (2011) stated that GA has been successfully used as a tool in computer programming, artificial intelligence, optimization and neural network training and information technology since its introduction by Holland (1975) to improve the performance of simple GA.

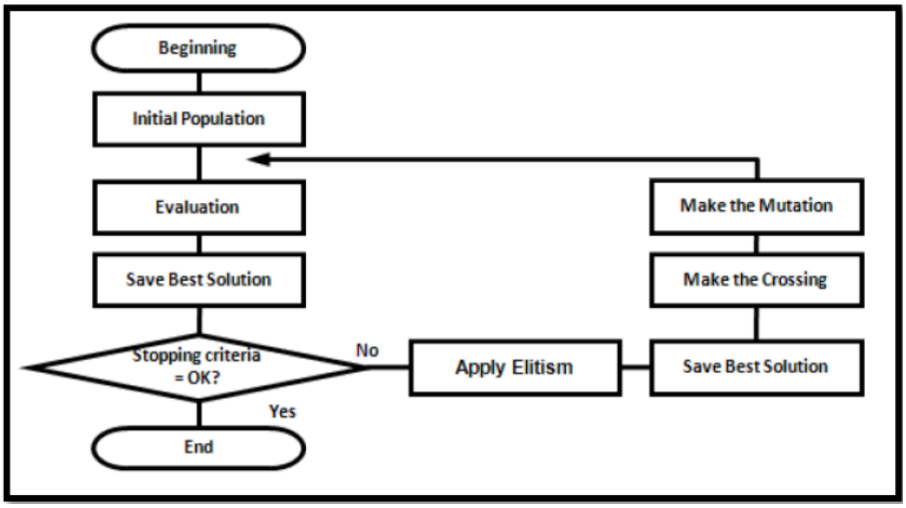

FIGURE 2 : Process Flow of Genetic Algorithm (Marco, et. al, 2012)

3.3 Optimization Procedure

3.3.1 Function Declaration

The objective function was first declared on the MATLAB so as to be solved by the GA Multi-objective Optimization tool to find optimal solution. Based on the system developed by Chen et al. (1995), three objectives were listed to be optimized. These objectives include the pump load/power output, f_1 (kW), overall efficiency, f_2 (%) and the fiscal savings, f_3 (USD). The design variables were;

i. Maximum operating pressure of the rankine cycle, x_a (MPa)
ii. Maximum operating temperature of the rankine cycle, x_b (K)
iii. Maximum solar collector temperature drop, x_c (K)

iv. The fluid flowrate of the rankine cycle, x_d (kg/s)

v. Ambient temperature, Z_a (K)

vi. Level of insolation, Z_b (K).

The system's formulation is shown below:

$$f_1 = -(24.947 + 16.011x_d + 1.306x_b + 0.820x_b x_d - 0.785Z_a - 0.497x_d Z_a + 0.228x_a x_b + 0.212x_a - 0.15x_b^2 + 0.13x_a x_d - 0.11x_a^2 - 0.0034x_b Z_a + 0.002x_a Z_a)10^{-3.24} \tag{1}$$

$$f_2 = -43.4783(0.18507 + 0.01041x_a + 0.0038Z_b - 0.00366Z_a - 0.0035x_c - 0.00157.x_b \tag{2}$$

$$f_3 = -(174695.73 + 112114.69x_d + 9133.8x_b + 5733.05x_b x_d - 5487.76Z_a - 3478.84x_d Z_a + 1586.48x_a x_b + 1486.84x_a - 1067.42x_b^2 + 916.26x_a x_d - 768.9x_a^2 - 242.88x_b Z_a + 152.4x_a Z_a)10^{-3.23} \tag{3}$$

Where the constraints are:

$$0.3 \le x_a \le 3 \,; 450 \le x_b \le 520 \,; 520 \le x_c \le 800 \,; 0.01 \le x_d \le 0.2 \,; 293 \le Z_a \le 303 \,; 800 \le Z_b \le 1000 \tag{4}$$

These formulation is translated to the MATLAB language format and written as follows:

```
f(1) = -(24.947+16.011*x(4)+1.306*x(2)+0.820*x(2)*x(4)-
0.785*x(5)+0.228*x(1)*x(2)*0.212*x(1)-0.15*x(2)^2+0.13*x(1)*x(4)-0.11*x(1)^2-
0.0034*x(2)*x(5)+0.002*x(1)*x(5))*10^(-3.24)

f(2) = -43.4783*(0.18507+0.01041*x(1)+0.0038*x(6)-0.00366*x(5)-0.0035*x(3)-
0.00157*x(2)

f(3) = -(174695.73+112114.69*x(4)+9133.8*x(2)+5733.05*x(2)*x(4)-5487.76*x(5)-
3478.84*x(4)*x(5)+1586.48*x(1)*x(2)+1486.84*x(1)-
1067.42*x(2)^2+916.26*x(1)*x(4)-768.9*x(1)^2-
242.88*x(2)*x(5)+152.4*x(1)*x(5))*10^(-3.23)
```

Where,

x_a = x(1)

x_b = x(2)

x_c = x(3)

x_d = x(4)

Z_a = x(5)

Z_b = x(6)

It was therefore declared in a new script and save as m-file with the name "multisolar.m" as in the FIGURE 3 below.

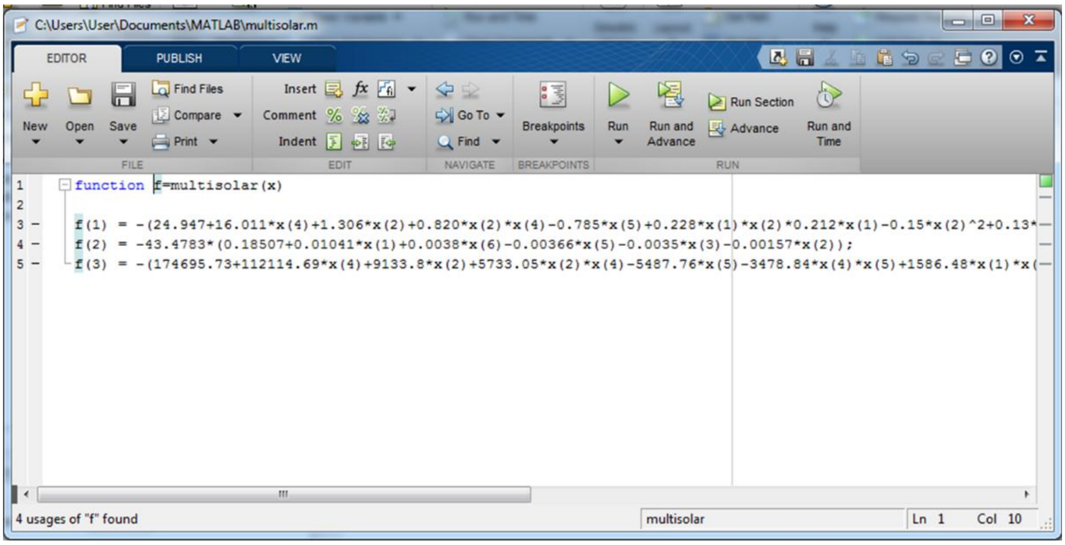

FIGURE 3: : MATLAB interface for function declaration

3.3.2 Genetic Algorithm Parameters

There are several parameters of genetic algorithm that are manipulated in order to get the best performance of the software however, not all the parameters are significant enough to affect the results. There were basically few GA parameters that were tuned in this project which are as explained below.

i. Population type

This specifies the type of the input to the fitness function. The population type can be set to be double vector or Bit string, or Custom. If custom is selected, creation, mutation, and crossover functions that work with the selected population type must be written. These functions must be specified in the fields Creation function, mutation function and Crossover function respectively.

ii. Population size

This specifies how many individuals are there in each generation. If population size is set to be a vector of length greater than 1, the algorithm creates multiple subpopulations. Each entry of the vector specifies the size of a subpopulation.

iii. Creation function

This specifies the function that creates the initial population. The constraint dependent default chooses:

- Uniform if there are no constraints
- Feasible population otherwise

Uniform creates a random initial population with a uniform distribution. Feasible population creates a random initial population that satisfies the bounds and linear constraints.

iv. Initial population

This specifies an initial population for the genetic algorithm. The default value is [], in which case GA uses the default Creation function to create an initial population. If a nonempty array in the Initial population field is entered, the array must have no more than Population size rows, and exactly Number of variables columns. In this case, the genetic algorithm calls a Creation function to generate the remaining individuals, if required.

v. Initial scores

This specifies initial scores for the initial population. The initial scores can also be partial. Do not specify initial scores with integer problems because GA overrides any choice you make.

vi. Initial range

This specifies the range of the vectors in the initial population that is generated by the GA creation uniform creation function. The Initial range is set to be a matrix with two rows and Number of variables columns, each column of which has the form [lb; ub], where lb is the lower bound and ub is the upper bound for the entries in that coordinate. If Initial range is specified to be a 2-by-1 vector, each entry is expanded to a constant row of length Number of variables. If an initial range is not specified, the default is [-10; 10] ([-1e4+1; 1e4+1] for integer-constrained problems), modified to match any existing bounds.

vii. Mutation Option

Mutation functions make small random changes in the individuals in the population, which provide genetic diversity and enable the genetic algorithm to search a broader space.

The mutation option by default chooses constraint dependent. Other options are:

- Gaussian if there are no constraints
- Uniform
- Adaptive feasible otherwise

Gaussian adds a random number to each vector entry of an individual. This random number is taken from a Gaussian distribution centred on zero. The standard deviation of this distribution can be controlled with two parameters; i.e. Scale and Shrink. The Scale parameter determines the standard deviation at the first generation. The Shrink parameter controls how standard deviation shrinks as generations go by. If the Shrink parameter is 0, the standard deviation is constant. If the Shrink parameter is 1, the standard deviation shrinks to 0 linearly as the last generation is reached.

Uniform is a two-step process. First, the algorithm selects a fraction of the vector entries of an individual for mutation, where each entry has the same probability as the mutation rate of being mutated. In the second step, the algorithm replaces each selected entry by a random number selected uniformly from the range for that entry.

Adaptive feasible randomly generates directions that are adaptive with respect to the last successful or unsuccessful generation. A step length is chosen along each direction so that linear constraints and bounds are satisfied.

viii. Crossover Options

Crossover options specify how the genetic algorithm combines two individuals, or parents, to form a crossover child for the next generation. Crossover function specifies the function that performs the crossover. There are few option in selecting the crossover function such as constraints dependent, scattered, single-point, two-point, intermediate, heuristic and arithmetic.

3.3.3 Genetic Algorithm Parameters Selection

The following parameters of genetic algorithm were randomly selected during the initial execution in the GA multi-objective optimization option tool before they were tuned.

- Population type: Double
- Population size: 200
- Creation function: constraint dependent
- Initial population: Default []
- Initial scores: Default []
- Initial Range: Default []
- Selection function: Tournament
- Reproduction Option: Crossover fraction of 0.8
- Mutation function: Constraint dependent
- Crossover function: Constraint dependent
- Fitness limit: -inf

3.4 Genetic Algorithm Optimization Flow

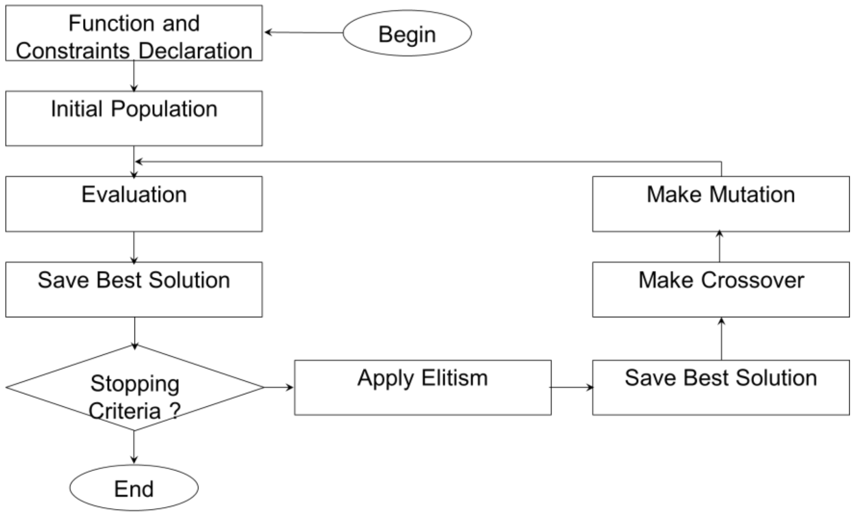

FIGURE 4: Genetic Algorithm Optimization Flow

3.5 Overall Methodological Flowchart

Figure 5 show the overall methodological flowchart. At the first stage, the function is declared inside the optimization tool of MATLAB followed by inputting the boundaries for the constraints from the solar irrigation model. Next, the search parameters are set based on the manipulation that is done through random search and observation. The initial search parameters are set as default based on the optimization presets. The optimization is run to get the set of pareto optimal solution. If the result is not satisfactory, the search parameters are modified to undergo new run of optimization. The fitness function data is recorded when the optimization result meet the satisfaction.

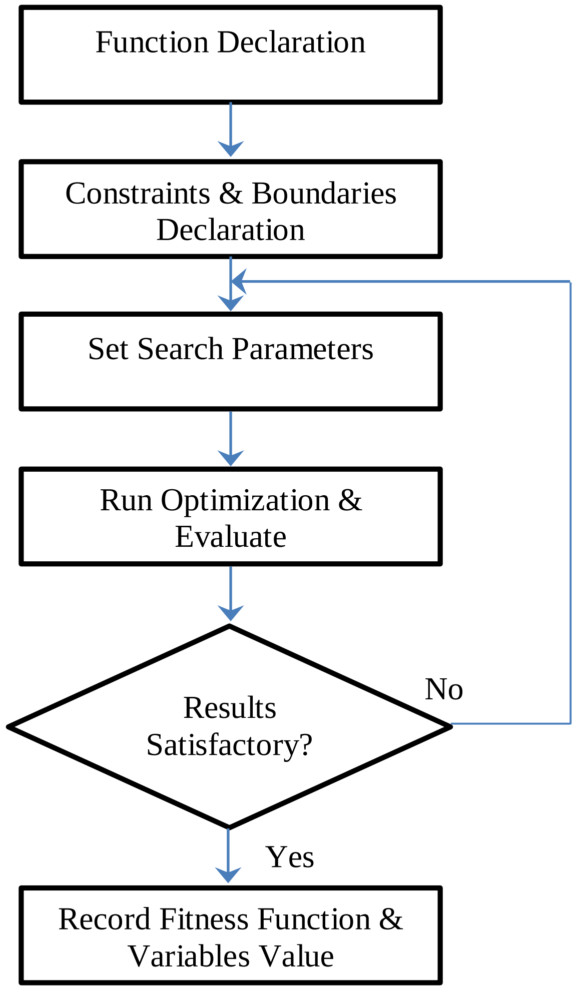

FIGURE 5: Overall Methodological Flowchart

3.6 Gantt Chart

TABLE 2: FYP I Gantt Chart

No.	Detail	Week													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Selection of project title														
2	Preliminary research work and proposal preparation														
3	Extended proposal submission														
4	Proposal defense														
5	Project work continues														
6	Submission of interim draft report														
7	Submission of final interim report														

TABLE 3: FYP II Gantt Chart

No.	Detail	Week														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Project work continues															
2	Submission of Progress Report															
3	Project work continues															
4	Pre-SEDEX															
5	Submission of Draft Report															
6	Submission of Dissertation															
7	Submission of Technical Paper															
8	Viva Oral Presentation															
9	Submission of Dissertation (Hard-Bound)															

3.7 Key Milestone

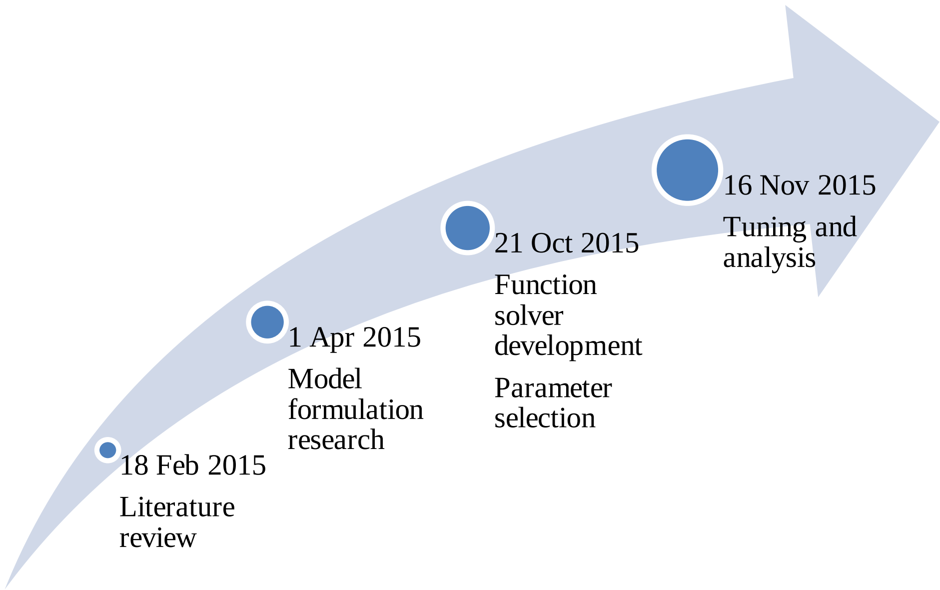

FIGURE 6: Key Milestone for FYP

CHAPTER 4

RESULT & DISCUSSION

4.1 Genetic Algorithm Parameter Screening

The search was initiated by using default setting. The main items which will be focused to be manipulated are the initial range and the stopping criterion which is fitness limit. The default initiating search parameters are as follows:

TABLE 4: Default Initiating Search Parameters

Population type	*Double*
Population size	*200*
Creation function	*Constraint dependent*
Initial population	*Default []*
Initial scores	*Default []*
Initial Range	*Default []*
Selection function	*Tournament*
Reproduction Option	*Crossover fraction of 0.8*
Mutation function	*Constraint dependent*
Crossover function	*Constraint dependent*
Fitness limit	*-inf*

The optimization is run to get the first set of solution for the specified formulation. For this run, the set of optimal solution obtained is as follows:

TABLE 5: Initial Solution after first simulation

f_1	17.3617
f_2	-17.2099
f_3	143043.8304
x_a	2.9992

x_b	450.0143
x_c	522.2156
x_d	0.1951
Z_a	293.0777
Z_b	996.4414

The aim of this optimization is to meet the targeted design specifications which the power output, f_1 must be approaching 20kW. For the efficiency, f_2 and fiscal savings, f_3, the must be maximized to be as close as possible to 20% and $150,000 respectively. However, the first run to get the optimal solution approaching the desired specification has encountered an error. The f_2 function value is not reasonable since it is negative. The expected value must be positive in order to make the first set of parameters reliable.

The initial range is then randomly manipulated to observe the changes on the solution. After some observation on the random results, the value is then decided to be specified at [1; 2], [1; 5] and [1; 10]. The search is run to get the following outcome:

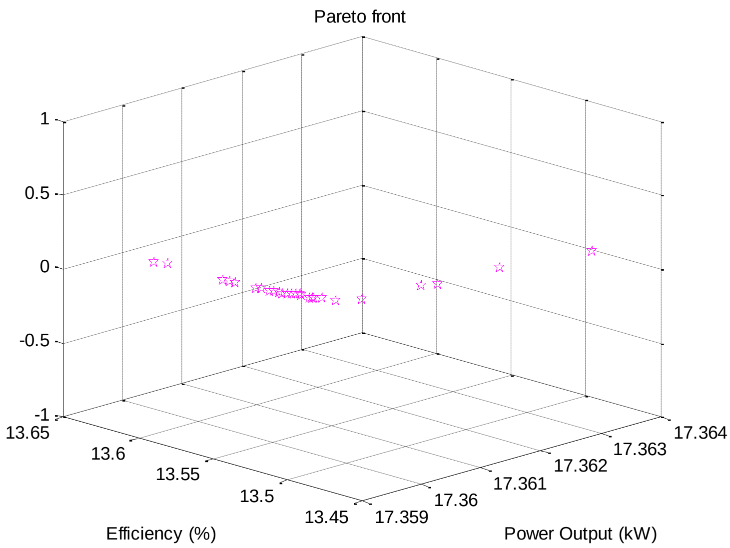

FIGURE 7: Pareto Front with initial range of [1; 2]

The FIGURE 7 above shows the pareto front for an initial range of [1; 2]. The value for f_1 is 17.3600 kW while f_2 and f_3 are 13.579 % and $143030.83 respectively. These set of parameter setting has shown an improvement from the initial search by making the f_2 value to become positive.

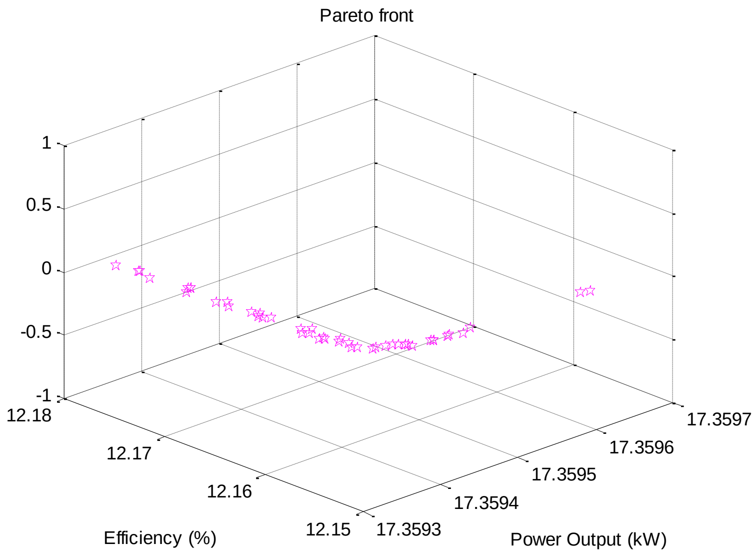

FIGURE 8: Pareto Front with initial range of [1; 5]

By changing the initial range to [1; 5], it can be observed from figure x that the efficiency has dropped from initial range [1; 2]. The set of values that is obtained from this setting are 17.3594 kW, 12.1527 % and $143025.91 for f_1, f_2 and f_3 respectively.

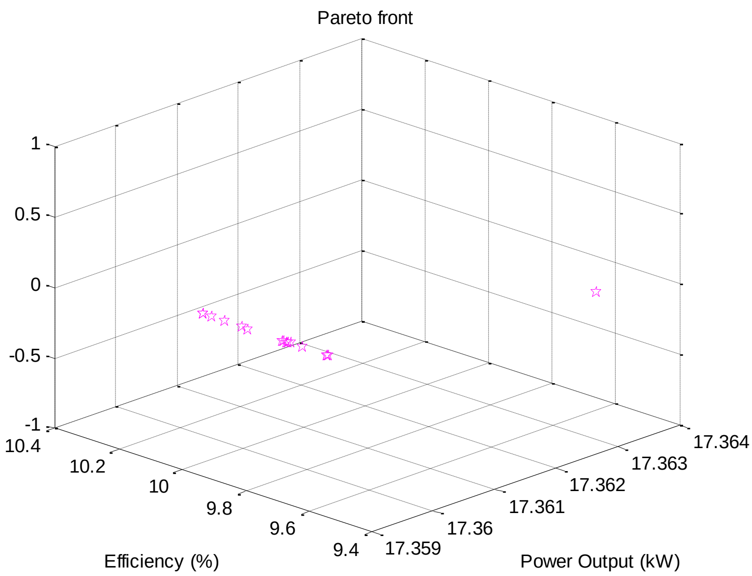

FIGURE 9: Pareto Front with initial range of [1; 10]

As the initial range increased to [1; 10], the efficiency continues to drop. In this run, the values obtained for all three objective functions are 17.3595 kW, 10.0323 % and $143026.27.

TABLE 6: Set of Optimal Solution with different Initial Range

	Initial Range			
	Default	[1; 2]	[1;5]	[1;10]
f_1	17.3617	17.3600	17.3594	17.3595
f_2	-17.2099	13.5794	12.1527	10.0323
f_3	143043.83	143030.83	143025.91	143026.27
x_a	2.9992	2.9985	2.9994	2.9990
x_b	450.0143	450.0063	450.0006	450.0015
x_c	522.2156	520.2053	520.0358	520.5433
x_d	0.1951	0.1996	0.1999	0.2000
Z_a	293.0777	293.0181	293.0053	293.0010
Z_b	996.4414	808.1743	816.6366	829.9354

After introducing a value of 10 on the fitness limit as the stopping criterion, there is a slight improvement on the value of f_2. The crossover fraction also varies the result obtained when the value is increased from 0.8 to 1.0.

4.2 Best, mean, and worst function values selection

TABLE 7 shows the best, mean and worst function after 3 run with different crossover fraction i.e. 0.6, 0.8, and 1.0. At this stage of optimization, the fitness limit of 10 is already introduced. The result is illustrated in FIGURE 10.

TABLE 7: Best, Mean, Worst function values after 3 run with different Crossover Fraction.

Crossover Fraction	Efficiency, f_2 (%)		
	Best	Mean	Worst
0.6	13.3524	13.35233	13.3523
0.8	14.4297	14.42963	14.4296
1.0	15.2355	15.2354	15.2352

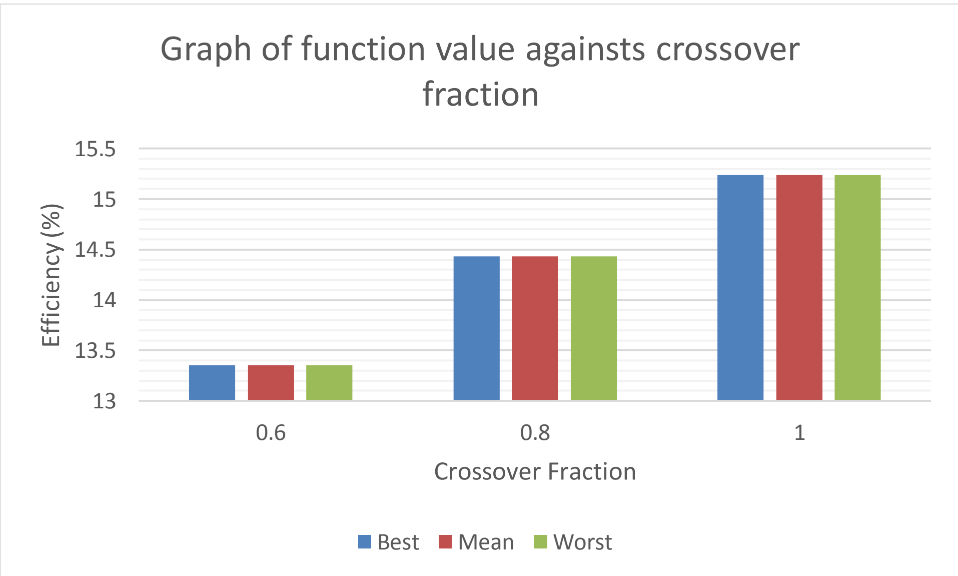

FIGURE 10: Graph of Function Values againsts Crossover Fraction

According to FIGURE 10, it can be clearly seen that increasing the crossover fraction until maximum value 1.0 will lead to a better efficiency of the system which give a mean value of 15.2354%. The trend gradually increasing from crossover fraction of 0.6 which only resulted in an efficiency of 13.35233%.

4.3 Best Parameter Tuning

After running few searches with different parameters and stopping criterion, it is identified that at this stage, the most value that can be resulted from initial range parameters is [1;2] as compared to [1;5] and [1;10]. In term of the crossover fraction, 1.0 has resulted the highest efficiency while stopping criterion (fitness limit) does not has significant difference between a value to another, but improved when an integer is introduced as compared to the default setting (-inf).

TABLE 8: Best Parameter Tuning

Population type	*Double*
Population size	*200*
Creation function	*Constraint dependent*
Initial population	*Default []*
Initial scores	*Default []*
Initial Range	*[1; 2]*
Selection function	*Tournament*
Reproduction Option	*Crossover fraction of 1.0*
Mutation function	*Constraint dependent*
Crossover function	*Constraint dependent*
Fitness limit	*10*

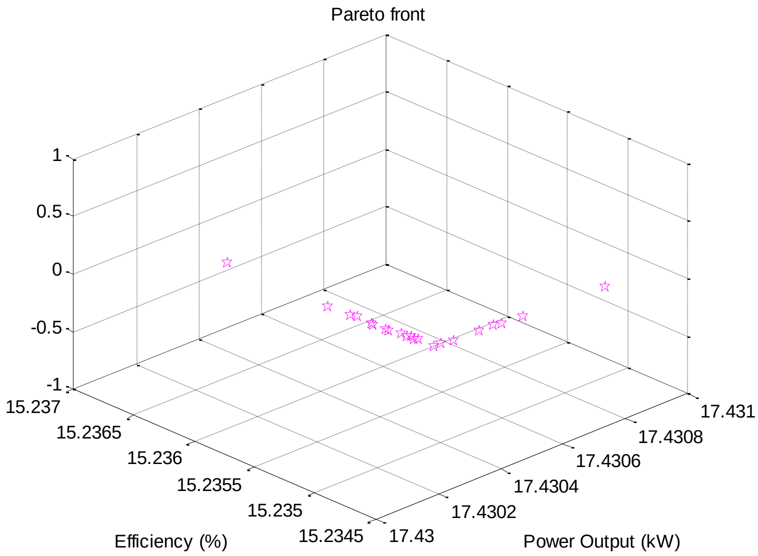

FIGURE 11: Pareto Front with Fitness Limit :10, Crossover Fraction: 1.0 and Initial Range: [1;2]

TABLE 8 shows the best set of parameter tuning while FIGURE 11 shows the pareto front with the best set of tuning. The set of pareto optimal solution with variables value is tabulated in TABLE 9 in comparison with other literatures.

4.4 Solution comparison between techniques

This genetic algorithm multi-objective optimization technique seems to be a common and easy approach in conducting the pareto optimal search for a complex engineering problem. However, it is found that the result is not relatively closed enough to the desired design objectives. The optimization is aiming to get power out maximized to 20kW while at the same time have the efficiency of 20% and fiscal savings approaching $150,000. One of the identified drawback of using the genetic algorithm metaheuristic technique is that the stopping criterion is not certain.

TABLE 9: Comparison between 3 search techniques

	Genetic Algorithm	Hyp-AP (Ganesan et al, 2014)	DSIDES (Chen et al, 1995)
Power Output, f_1(kW)	17.4303	20.7987	20.003
Efficiency, f_2 (%)	15.2355	17.4495	19.45
Fiscal Savings, f_3 (USD)	143533.10	148927	141143
[redacted]			
Maximum Press., x_a (MPa)	1.9714	0.6206	3
Maximum Temp., x_b (K)	450.0682	456.5	450
Solar Collector Temp., x_c (K)	520.1509	524.661	550
Fluid Flowrate, x_d (kg/s)	0.1927	0.038835	0.0258
Ambient Temp., Z_a (K)	293.0298	302.707	0.02577
Level of Insolation, Z_b (K)	800.9515	807.545	0.02577

CHAPTER 5
CONCLUSION & RECOMMENDATION

The main objectives of doing this research is to identify one of the complex engineering problem which is related to the agricultural which nowadays has entered the sustainable energy development domain which embed sources of renewable energy like solar into the design of the system. One of the system configuration of this irrigation system is by combining the solar collector heat cycle with Rankine cycles 4 devices. Conventional way of supplying heat source to the boiler by using fuel is no longer feasible in a long run due to current fuel price fluctuation. To optimize this complex problem, suitable technique should be embedded. Ganesan et al. (2013) and Chen et al. (1994) have tried different approach in finding the non-dominated solution which will satisfy the design objectives by using Analytical Programming and DSIDES respectively.

This paper has introduced another different method to get the optimal set of solution. In order to get the best parameter setting, tuning has been done. It is identified that for constraint dependent mutation and crossover, the best initial range is [1;2] while the fitness limit is 10 as for the stopping criterion. The most maximized value if the objectives can be obtained if the crossover functions is set 1.0 instead of the default value of 0.8. At this stage of simulation, the optimal set of solution has led to these set of data:

TABLE 10: Set of Optimal Solution by using Genetic Algorithm

Fitness Function	Power Output, f_1(kW)	17.4303
	Efficiency, f_2 (%)	15.2355
	Fiscal Savings, f_3 (USD)	143533.10

Variables and Noise Factor	Maximum Press., x_a (MPa)	1.9714
	Maximum Temp., x_b (K)	450.0682
	Solar Collector Temp., x_c (K)	520.1509
	Fluid Flowrate, x_d (kg/s)	0.1927
	Ambient Temp., Z_a (K)	293.0298
	Level of Insolation, Z_b (K)	800.9515

In future works, it is recommended that other meta-heuristic algorithms such as Genetic Programming (Koza, 1992) and other hybrid algorithms should be embedded with this solar irrigation system. These approaches should then be tested with other industrial application problems.

REFERENCES

Enitan, Abimbola M., & Adeyemo, J. (2013). Food processing optimization using evolutionary algorithms. *African Journal of Biotechnology*, *10*(72), 16120-16127.

Amouzgar, K. (2012). Multi-objective optimization using Genetic Algorithms.

Beasley, D., Martin, R., & Bull, D. (1993). An overview of genetic algorithms: Part 1. Fundamentals. *University computing, 15*, 58-58.

Belegundu, A., Murthy, D., Salagame, R., & Constants, E. (1994). *Multi-objective optimization of laminated ceramic composites using genetic algorithms.* Paper presented at the A Collection of Papers, 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization.

Campana, P. E., Li, H., Zhang, J., Zhang, R., Liu, J., & Yan, J. (2015). Economic optimization of photovoltaic water pumping systems for irrigation. *Energy Conversion and Management, 95*(0), 32-41.

Chen, W., Tsui, K.-L., Allen, J. K., & Mistree, F. (1994). Integration of the response surface methodology with the compromise decision support problem in developing a general robust design procedure.

Deb, K., & Beyer, H.-G. (2001). Self-adaptive genetic algorithms with simulated binary crossover. *Evolutionary Computation*, *9*(2), 197-221.

Freeman, J., Hellgardt, K., & Markides, C. N. (2015). An assessment of solar-powered organic Rankine cycle systems for combined heating and power in UK domestic applications. *Applied Energy, 138*(0), 605-620.

Ganesan, T., Elamvazuthi, I., Shaari, K. Z. K., & Vasant, P. (2013). Hypervolume-driven analytical programming for solar-powered irrigation system optimization *Nostradamus 2013: Prediction, Modeling and Analysis of Complex Systems* (pp. 147-154): Springer.

Gao, X., Liu, J., Zhang, J., Yan, J., Bao, S., Xu, H., & Qin, T. (2013). Feasibility evaluation of solar photovoltaic pumping irrigation system based on analysis of dynamic variation of groundwater table. *Applied Energy, 105*(0), 182-193.

Garcia-Najera, A., & Bullinaria, J. A. (2009). *Bi-objective optimization for the vehicle routing problem with time windows: Using route similarity to enhance performance.* Paper presented at the Evolutionary Multi-Criterion Optimization.

Kelley, L. C., Gilbertson, E., Sheikh, A., Eppinger, S. D., & Dubowsky, S. (2010). On the feasibility of solar-powered irrigation. *Renewable and Sustainable Energy Reviews, 14*(9), 2669-2682.

Konak, A., Coit, D. W., & Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety, 91*(9), 992-1007.

Marler, R. T., & Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization, 26*(6), 369-395.

Mitchell, M. (1995). Genetic algorithms: An overview. *Complexity, 1*(1), 31-39.

Schaumann, E. J., Balling, R. J., & Day, K. (1998). *Genetic algorithms with multiple objectives.* Paper presented at the Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization.

Senol, R. (2012). An analysis of solar energy and irrigation systems in Turkey. *Energy Policy, 47*(0), 478-486.

Wang, J., Yan, Z., Zhao, P., & Dai, Y. (2014). Off-design performance analysis of a solar-powered organic Rankine cycle. *Energy Conversion and Management, 80*(0), 150-157.