Evaluation of Evolutionary Algorithms for Multi-objective Train Schedule Optimization

C.S. Chang and Chung Min Kwan

National University of Singapore Department of Electrical and Computer Engineering,
10 Kent Ridge Road, Singapore 119260
{eleccs, g0301034}@nus.edu.sg
http://www.nus.edu.sg

Abstract. Evolutionary computation techniques have been used widely to solve various optimization and learning problems. This paper describes the application of evolutionary computation techniques to a real world complex train schedule multiobjective problem. Three established algorithms (Genetic Algorithm GA, Particle Swarm Optimization PSO, and Differential Evolution DE) were proposed to solve the scheduling problem. Comparative studies were done on various performance indices. Simulation results are presented which demonstrates that DE is the best approach for this scheduling problem.

1 Introduction

The problem of minimizing operating costs and maximizing passenger comfort of a medium sized mass rapid transit (MRT) system is multiobjective and conflicting. It is affected by factors such as the dispatch frequency of trains; dwell time at passenger stations, waiting time for passengers as well as how smooth a train travels. Recently, evolutionary algorithms were found to be useful for solving multiobjective problems (Zitzler and Thiele 1999) as it has some advantages over traditional Operational Research (OR) techniques. For example, considerations for convexity, concavity, and/or continuity of functions are not necessary in evolutionary computation, whereas, they form a real concern in traditional OR techniques. In multiobjective optimization problems, 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 (dominate) them when all objectives are simultaneously considered. They are known as pareto-optimal solutions. Pareto-optimality is expected to provide flexibility for the human decision maker.

In this paper, three pareto-based approaches for solving the multiobjective train scheduling problem were investigated. The paper is organized as follows: The formulation of the problem is detailed in section 2 followed by the proposed algorithms in section 3. Experiments are then presented in section 4 and conclusions are drawn in section 5.

G.I. Webb and Xinghuo Yu (Eds.): AI 2004, LNAI 3339, pp. 803–815, 2004.

2 Formulation of Problem

2.1 Motivation

Modern MRT systems are concerned with providing a good quality of service to commuters without compromising safety. An important assessment on the quality of service is the comfort level of the passengers. Besides the quality of service, decision makers of MRT system are also concerned with minimizing the operating costs of running the system. However, under normal conditions operating costs could be minimized only with passenger comfort level compromised. There is hence a need to find the optimal set of solutions for these conflicting objectives.

In our work a spreadsheet database was created for the study of various train parameters and their effect on operating costs. The database enables key parameters to be adjusted, and supports a framework for generating a predetermined train schedule. The set of train schedules can subsequently be fed into an algorithm known as Automated Train Regulator (ATR) [1] for dynamic simulation of the schedule, fine tuning, and the study of possible conflicts encountered during implementation. The piece of work seeks to automate the process of optimizing the predetermined timetable by employing evolutionary algorithms; where the schedule is previously tuned manually. The process is extended to incorporate a simple passenger discomfort function to demonstrate the feasibility of the algorithms in the multiobjective case. The whole task is simplified by making some assumptions. a.) Only key variables affecting the operating costs and passenger comfort level are considered. b.) Certain fixed costs like the salary of the train drivers, transport allowances, the number of working hours per day etc are fixed in advance. c.) Passenger flows during different periods of a working or festive day are also known in advance.

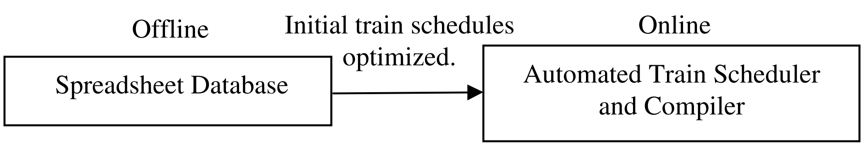

Fig. 1. Spreadsheet and Scheduler Relationship

2.2 Key Variables and Constraints

The following equations define the basic relationship of the variables affecting the multi-objective optimization problem. Essentially the operating cost increases with the increase of the number of trains T in the system.

$$\text{No. of trains T} = \text{Cycle Time / Headway} \ . \tag{1}$$

$$\text{Cycle Time} = \text{Run Time} + \text{Layover Time} + \text{Dwell Time} \ . \tag{2}$$

The passenger comfort level is defined to be affected by the headway in a simple model:

$$\text{Passenger Comfort Level} = \alpha\,(\text{headway})^2. \tag{3}$$

Where the headway, run time, layover time and dwell time in the above equations are identified as key variables to be optimized. An explanation for the key variables is provided below.

Headway. This is defined as the distance (time gap) between two successive trains traveling in the same direction on the same track.

Dwell Time. This is part of the cycle time taken by a train to stop at a particular station. This key variable is affected by passenger flow and the times taken by passengers to board/ alight each train.

Run Time and Coast Level. The run time of a train is defined as the time taken for its journey to travel between stations. Coast level describes the train movement profile and the amount of energy usage. As illustrated in a train velocity-time profile between adjacent stations (Fig. 2), the train is accelerated after departure from the first station. When it reaches its desired velocity, the train can either (**a**) remain powering (Coast Level 0) to cover a given distance within the shortest time but requiring the highest energy consumption; (**b**) turn the motive power on and off at certain desired velocities (Coast Level 1), or (**c**) turn off the motive power completely (Coast Level 2) for the longest run time but consuming the lowest energy.

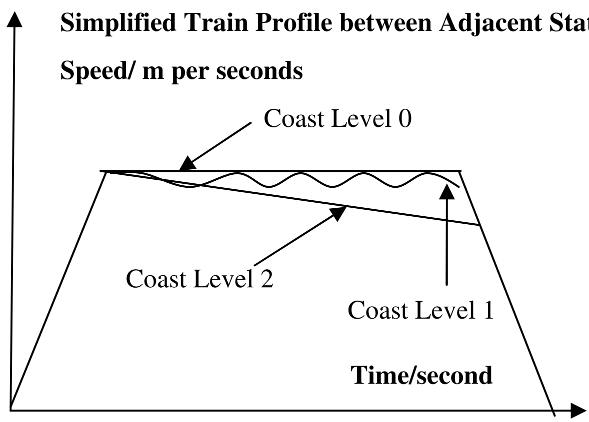

Fig. 2. Train Profile and Coast Level

Layover Time. This is the time taken for a train to reverse its direction at a terminal station, which includes the time taken for changing the train driver and completing other procedures required at the terminal station.

Hard Constraints (Safety). These are associated with the safe running of trains, passenger safety and proper train functioning. (Cannot be violated)

- Minimum headway – this must be observed at all times to ensure a safe distance between a train from the train immediately ahead/behind
- Minimum routine inspection time at terminal stations – minimum time involved at terminal stations for routine inspections of trains to ensure a safe journey
- Minimum dwell time – ensures that trains stay long enough at each station.
- Speed limits – limits the train speed at track locations of high curvature or entry/ exit points of underground/ over-ground sections
- Maximum allowable passenger loading –limits the maximum number of passengers allowed in a train

Soft Constraints (Quality of Service). These are associated with the quality of service and passenger comfort, which include:

- Policy Headway – This is the maximum allowable headway set at peak hours to ensure that passengers would be able to receive a minimum quality of service.

2.3 Objective Function

The objective function of our train scheduling problem is to minimize the operation cost and passenger discomfort level, subjected to the various constraints in section 2.2. The problem statement is stated as follows:

Minimize $f(x) = f_1(x), f_2(x)$ where $f_1(x)$ = Operating costs, $f_2(x)$ = Passenger Discomfort level (Subjected to Safety and Quality of service constraints). (4)

where the passenger discomfort level is the inverse of passenger comfort level as defined in equation (3), and both objectives are conflicting.

3 Optimization Methods

The multi-objective train scheduling problem as defined in previous section is a complex problem which could not be solved by conventional OR techniques. The use of the spreadsheet database allows evolutionary algorithms to be employed elegantly; avoiding the complexities of the mathematical equations governing the objective functions as these are handled by the spreadsheet itself. Three techniques are adopted to solve the multi-objective train scheduling problem, GA, PSO and DE. PSO strictly belongs to a special class known as swarm optimization that is inspired by the choreography of a bird flock; it is included in our analysis to provide interesting alternative for comparison with the conventional evolutionary algorithms.

Pareto-based approach (Goldberg) [2] is preferred and adopted in this work because it is able to provide a whole range of non-dominated solutions, providing flexibility for the decision maker to select and adopt the best solution as he/she deemed fit. Non-pareto methods like the weighted sum (Horn 1997) [3] is difficult to implement in this work because there is no common metric between the different criteria, leading to difficulties in setting a suitable weight.

3.1 Multi-objective Evolutionary Computation Algorithm

Many well known pareto-based algorithms are available to be adopted in solving the train-scheduling problem. [4,11,13]. For comparison purposes, care must be taken to ensure that the algorithms are as generic as possible and could be used for all three algorithms without providing unfair advantage. The generic multiobjective algorithm adopted in our work is listed as follows (modified from SPEA, [4])

1. Generate an initial population P and create the empty external nondominated set $P^{/}$(Pareto front set).

2. Copy the nondominated members of P into $P^{/}$.

3. Remove solutions within $P^{/}$ which are covered by any other member of $P^{/}$.

4. If the number of externally stored nondominated solutions exceeds a given maximum $N^{/}$, prune $P^{/}$ by means of clustering. ($N^{/}$ = Pruned Pareto front set)

5. Calculate the fitness of each individual in P as well as in $P^{/}$. (Fitness evaluation)

6. Apply (**GA, PSO or DE**) to the problem.

7. If the maximum number is reached, then stop, else go to step 2.

3.2 Methodology and Design of Proposed Solutions

In the comparative study between the three algorithms proposed, the main challenge is to ensure that algorithms could be compared in a meaningful and fair way. Two design steps were adopted to achieve it. The first step involves studying the three algorithms and identifying the parameters to be kept constant to ensure a fair comparison. Subsequently, the second stage involves preliminary studies to tune the set of control parameters for each proposed algorithm.

3.2.1 Techniques and Identification of Control Parameters

Most common evolutionary computation techniques follow roughly the same process [5]: 1.)Encoding of solutions to the problem as a chromosome; 2.) A function to evaluate the fitness, or survival strength of individuals; 3.) Initialization of the initial population; selection operators; and reproduction operators. For the three algorithms proposed, the components are listed as follows:

Table 1. Main components for the various algorithms

	GA	PSO	DE
Encoding	Real	Real	Real
Selection	Roulette Wheel	See equation 5 and 6 [8,9]	See equation 7 and 8 [10,11]
Reproduction	Mutation, Crossover [6,7]		
Population size	30	30	30

The various equations as listed in the tables are as follows:

$$\underline{v}_i = w^* \, \underline{v}_i + c1 * rand1 * (\underline{pbest}_i - \underline{present}_i) + c2 * rand2 * (gbest - \underline{present}_i). \tag{5}$$

$$\underline{present}_i = \underline{present}_i + \lambda^* \, \underline{v}_i \, . \tag{6}$$

Equations (5) and (6) are modifying equations for the PSO algorithm. The adjustments of the individual candidate solutions are analogous to the use of a crossover operator in conventional evolutionary computation method. $\underline{v}_i$ represents the velocity vector of the *i*th particular candidate solution of the population; $\underline{pbest}_i$ is the *i*th candidate solution's best experience attained thus far; $\underline{present}_i$ is the current position along the search space of the *i*th candidate; rand1 and rand2 are random terms in the range from 0 to 1; gbest represents the best candidate solution in the particular generation and c1, c2, w and are weight terms. The detailed workings could be found in [8,9].

$$\underline{v}_i = \underline{x}_{i,G} + \lambda.(\underline{x}_{best,G} - \underline{x}_{r1,G}) + F.(\underline{x}_{r2,G} - \underline{x}_{r3,G}) \; . \tag{7}$$

$$\underline{u}_j = \underline{v}_j \;\; \text{for} \;\; j = \langle n \rangle_D, \langle n+1 \rangle_D, \ldots\ldots \langle n+L-1 \rangle_D \tag{8}$$

$$\underline{u}_j = \underline{x}_j \quad \text{otherwise}$$

In the DE process, equations (7) and (8) detail the population modification process. Equation (7) is analogous to the mutation process while equation (8) is analogous to the crossover process in conventional evolutionary algorithms. The term $\underline{v}_i$ in equation (7) represents the *i*th trial vector; $\underline{x}_{i,G}$ is the *i*th candidate solution of the current generation G; $\underline{x}_{best.G}$ is the best candidate solution of the current generation; $\underline{x}_{r2,G}$ and $\underline{x}_{r3,G}$ are randomly selected candidate solutions within the population size; λ and F are weight terms and finally the term $\underline{u}_j$ represents the *j*th element that has gone through the process of crossover based on a crossover probability of CR. The detailed workings could be found in [10,11].

In our comparative studies, it is considered a common practice that they must have the same representation, operators, and parameters to ensure a fair comparison. In this part of work the population size (30), coding type (Real) and the method of initializing the population size (random) are identified as components to be kept constant. The other control parameters relating to the three algorithms (Mutation, Crossover for GA; C1, C2, w, λ for PSO; λ, F, crossover for DE) are allowed to be tuned.

The next part presents the preliminary studies done to tune the various control parameters.

3.2.2 Tuning and Customization of Algorithms

The control parameters for each algorithm were determined through extensive experimentation. The tuned control parameters are summarized as follows:

Table 2. Tuned Control Parameters for GA, PSO, DE

Parameter for EA	GA	PSO	DE
Population size	30	30	30
Coding Type	Real number	Real number	Real number
Selection and Reproduction mechanisms	0.8 (One-point **crossover**)	2 (**C1** & **C2**)	0.5 to 0 (linear decrease,λ)
	Roulette Wheel (**Selection**)	4.0 to 2.0 (linear decrease, **w**)	4.0 to 0 (linear decrease, **F**)
	Mutation rate Decreases linearly (starts with 0.02, decreases to 0)	0.8 (λ)	0.3 (**crossover**)

Many available techniques for controlling and selecting the parameters for a general multi-objective evolutionary algorithm are discussed [5]. In our work however, simple forms of dynamic tuning were used in all three algorithms and it is deemed sufficient for our application based on the satisfactory results obtained for the single unconstrained optimization case. The rationale of incorporating such a variation was due to its simplicity and fast implementation. It allows for a large exploration of the search space at the initial stage (global exploration) and a faster convergence after sufficient iterations (local fine tuning). Simplicity in implementation will facilitate customization of the algorithms in future for formulating a more comprehensive way to control the parameters, which will lead to generalizability and customization. Further work may lead to adaptive control and other forms of more complex control for parameter tuning.

4 Simulation Results and Analysis

The East-West line of a typical medium sized MRT system was used for testing the feasibility of the developed algorithms. The test system consists of 26 passenger stations studied at morning peak. All simulations were carried out on the same

Pentium 4 PC (2.4 GHz). 30 independent runs were performed per algorithm to restrict the influence of random effects. Initial population was the same for each run for the different algorithm. The termination condition for each simulation is 500 iterations. Analysis is performed in two stages—the first stage consists of testing the three algorithms on the single objective of operating costs aimed at verifying the feasibility of the algorithms. Subsequently the second stage solves the complete multi-objective train scheduling problem. The computer program for the optimization process is written in Microsoft Visual C#, under the .NET framework.

4.1 Stage 1—Operating Cost as SINGLE Objective

To gain insight and confidence in the three proposed algorithms, simulations were performed for operating costs as a single objective (Passenger comfort level omitted) with the optimal known in advance as the optimization processes were run under no constraints. This stage is necessary to ensure the workability of the proposed algorithms before embarking on the full multiobjective optimization. The range of variables and their expected optimized values are listed as follows:

Table 3. Variables for optimization *(UP and DOWN directions)*

Variables	Allowable Range	Expected Optimal
Headway	60s to 180s	180s
Total Dwell time	540s to 1080s	540s
Layover time	120s to 240s	120s
Coast level	0 to 2	2

The results for each algorithm converged to their expected optimized values as shown in Table 6 and Fig. 3. Referring to the results, it was noted clearly that GA falls behind PSO and DE in performance. In terms of the amount of time taken to run the optimization process GA clocked a time of 14 minutes, which is about 16% slower than PSO and 27% slower than DE. Moreover, GA was not able to produce as good a result as either PSO or DE. It is concluded at this stage that GA is inferior to both PSO and DE.

There is a close contention between PSO and DE. While DE takes a faster time (11minutes) to complete the simulation, with PSO (12 minutes) lagging by 9%, PSO converges at a faster rate compared to DE (shown in Fig. 7.) At this stage therefore, it is not clear which is the best algorithm for our application. However, the feasibility of all the algorithms have been demonstrated through this stage; as all are shown to be capable for the single objective train operation costs optimization.

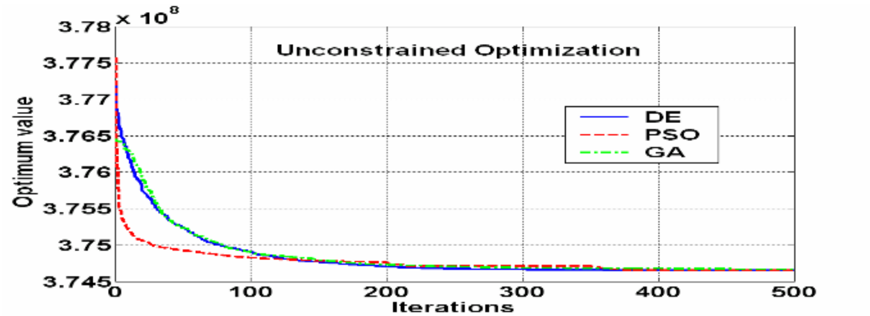

Fig. 3. Results for Unconstrained Optimization (*Averaged over 30 runs*)

Table 4. Results and Comparison (*Average Values over 30 runs*)

Type of comparison	GA	PSO	DE
Average Time taken	14mins	12mins	11mins
Convergence	>500 iterations	About 280 iterations	About 320 iterations
Final Headway West Bound	179	179	179
Final Headway East Bound	179	179	179
Final Dwell time West Bound	540	540	540
Final Dwell time East Bound	542	540	540
Final Layover time West Bound	120	120	120
Final Layover time East Bound	122	120	120
Final Coast level	2	2	2
Lowest Cost Achieved (Testing Values)	374040350	374035708	374035708

4.2 Stage 2 – Multi-objective Optimization

Having gained the initial confidence on the feasibility of the algorithms, the challenge in this stage is to determine the optimal set of solutions for the multiobjective problem defined in equation (4). Based on the passenger comfort level defined in equation (3), the parameter α is set to be 0.01. The termination condition for the multiobjective case has been set to 800 iterations to avoid premature termination. Population size (N) and the maximum number of externally stored nondominated solutions (N′) was set to 30. Control parameters of the three algorithms were kept as determined earlier.

A random search algorithm (Rand) was added to serve as an additional point of reference [4, 12]. This probabilistic algorithm was performed with the GA algorithm without any selection, mutation and crossover.

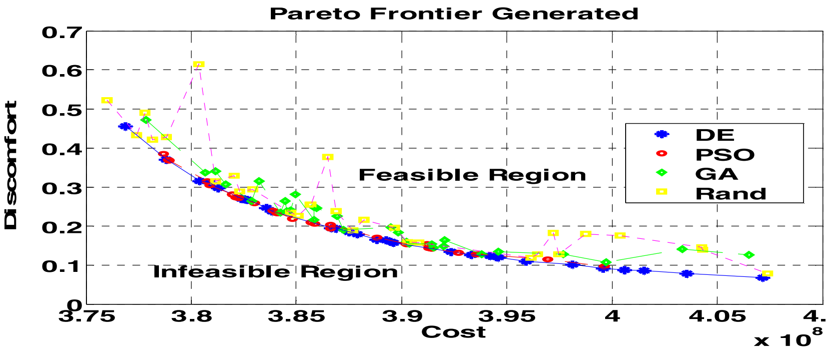

Fig. 4. Best Pareto-Front Generated for the Multi-Objective problem (*30 runs,N and N′ are both set to 30*)

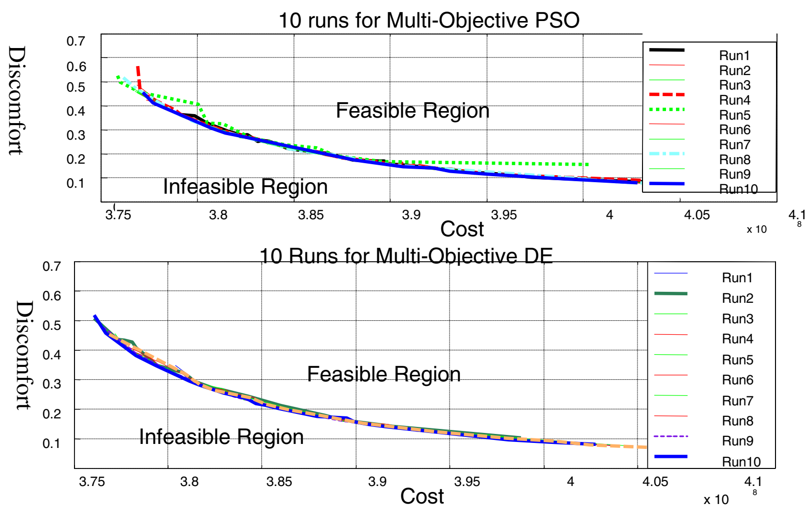

Fig. 5. 10 Independent runs for PSO and DE (*GA and Rand omitted as they are shown to be inferior to PSO and DE*)

To provide a fair comparison in the multiobjective scenario, certain concepts in the MOPSO method [13] was adopted and the global best term in the PSO update equations (the term gbest) was replaced by a repository storage used to store the positions of the particles that represent nondominated vectors. In terms of DE, an alternative form of equation (7) was used (Storn and Price, Scheme DE 1) which does not take into account $\underline{x}_{best,G}$. $\underline{x}_{i,G}$ has also been replaced by the term $\underline{x}_{r1,G}$, a randomly selected candidate within the population size. The results for the simulations are as follows:

The results were obtained by running 30 independent runs and plotting the best non-dominated front extracted for each algorithm over the 30 runs. In an additional plot (Fig. 5) the figures showing 10 independent runs for two algorithms provide an additional insight into the problem. It was noted that the multi-objective DE and multi-objective PSO were able to maintain consistent results showing minimal deviation (Except one for PSO).

Table 5. Results of Multiobjective Optimization Based on number of points (A) is dominated by (B) (*30 runs, best results taken*)

(A)	**No. of solution points dominated by (B)**				
	Rand	**GA**	**PSO**	**DE**	**Mean**
Rand	--	11	27	28	22
GA	3	--	20	20	14.3
PSO	0	0	--	0	0
DE	0	0	0	--	0

A useful quantitative measure adopted from [4, 12 coverage] was used to aid in the evaluation process. It displays the number of solution points generated by an algorithm (A) that is dominated by another algorithm (B). (See Table 5) The higher the mean demonstrates that the more that particular type of algorithm is dominated by others. From the results it is seen that Rand is most dominated by other algorithms (as expected) while GA is the second worse. DE and PSO are not dominated by each other (with a mean of 0) However, it was noted that DE was able to provide a better spread of solutions compared to PSO. Based on the qualitative process of spread as well as the consistency of results demonstrated in Fig. 5, DE was shown to perform better in this piece of work.

5 Discussion

From the results presented certain points are drawn. Firstly, evolutionary algorithms are shown to work well with the single objective of operation costs in section 4.1, where the three algorithms presented were able to effectively reach the unconstrained optimized values. The successful results demonstrated in section 4.1 allowed section 4.2 to be carried out meaningfully (else there would be no basis of multiobjective optimization if the single case failed). While not clearly observable in the single objective case, DE has been shown superior based on the two performance criteria presented in section 4.2. We would therefore draw the conclusion that DE overall outperforms the other two algorithms. While experimentally this is shown, the authors have not been able to define mathematically why DE is superior to others in this type of problems. Future work seeks to expand the complexity of the problem (by bringing in more objectives, dropping certain assumptions) as well as bring in other forms of evolutionary algorithms for more extensive testing.

6 Conclusion

This study compared three evolutionary algorithms on a multiobjective train scheduling problem. By breaking up the analysis into two stages, we seek to first verify the feasibility of the algorithms in the single objective of operating costs and subsequently extending the work to the multiobjective case of operating costs and passenger comfort. The capabilities of all the three algorithms have been demonstrated in both the stages, with DE showing remarkable performance. Future work would seek to include the use of evolutionary algorithms in more complex mass rapid transit planning operations.

References

1. C.S. Chang, Chua C.S., H.B. Quek, X.Y. Xu, S.L., Ho, "Development of train movement simulator for analysis and optimization of railway signaling systems", *Proceeding of IEE conference on development of Mass Transit Systems, 1998, pp. 243-248.*
2. D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
3. J. Horn, "F1.9 Multicriteria decision making," in *Handbook of Evolutionary Computation,* T. Back, D.B. Fogel, and Z. MIchalewicz, Eds. Bristol, U.K.: Inst. Phys. Pub., 1997
4. E. Zitzler, L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach", *IEEE Trans. on Evolutionary Computation,* vol. 3, No. 4, Nov 1999
5. Engelbrecht, A.P., "Computational Intelligence: An Introduction," Hoboken, N.J.: J. Wiley & Sons, c2002.
6. C.S. Chang, W.Q. Jiang, S. Elangovan, "Applications of Genetic Algorithms to determine worst-case switching overvoltage of MRT systems", *IEEE Proc- electrPowerApp,* vol. 146, No.1, Jan 1999.
7. J.X. Xu, C.S.Chang, X.Y. Wang, "Constrained multiobjective global optimization of longitudinal interconnected power system by Genetic Algorithm", IEE Proc Gener.Transm..Distrib.,Vol. 143. No.5, Spetember 1996, pp. 435-446
8. J. Kennedy, and R.C. Eberhart. "Swarm Intelligence", Morgan Kaufmann Publishers, 2001
9. (Particle Swarm Optimization Tutorial), X. Hu, Available: http://web.ics.purdue.edu/~hux/ tutorials.shtml
10. R. Storn, K. Price, "Differential Evolution—A simple and efficient adaptive scheme for global optimization over continuous space", Technical Report TR-95-012, ICSI

11. C.S. Chang, D.Y. Xu, and H.B. Quek, "Pareto-optimal set based multiobjective tuning of fuzzy automatic train operation for mass transit system", IEE Proc-Electr. Power Appl., Vol. 146, No. 5, pp. September 1999, pp 577-587
12. E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms – A comparative case study," in *5th Int. Conf. Parallel Problem Solving from Nature (PPSNV),* A.E. Eiben, T. Back, M. Schoenauer, and H.-P. Schwefel, Eds. Berlin, Germany: Springer-Verlag, 1998, pp. 292-301.
13. C.A. Coello Coello, G. Toscano, M.S. Lechuga, "Handling Multiple Objectives with Particle Swarm Optimization," *IEEE Trans. on Evolutionary Computation,* vol. 8, No. 3, June 2004