

Real-Coded ECGA for Economic Dispatch

Chao-Hong Chen
Dept. of Computer Science
National Chiao Tung University
HsinChu City 300, Taiwan
chchen@nclab.tw

Ying-ping Chen
Dept. of Computer Science
National Chiao Tung University
HsinChu City 300, Taiwan
ypchen@nclab.tw

ABSTRACT

In this paper, we propose a new approach that consists of the extended compact genetic algorithm (ECGA) and split-on-demand (SoD), an adaptive discretization technique, to economic dispatch (ED) problems with nonsmooth cost functions. ECGA is designed for handling problems with decision variables of the discrete type, while the decision variables of ED problems are oftentimes real numbers. Thus, in order to employ ECGA to tackle ED problems, SoD is utilized for discretizing the continuous decision variables and works as the interface between ECGA and the ED problem. Furthermore, ED problems in practice are usually hard for traditional mathematical programming methodologies because of the equality and inequality constraints. Hence, in addition to integrating ECGA and SoD, in this study, we devise a repair operator specifically for making the infeasible solutions to satisfy the equality constraint. To examine the performance and effectiveness, we apply the proposed framework to two different-sized ED problems with nonsmooth cost function considering the valve-point effects. The experimental results are compared to those obtained by various evolutionary algorithms and demonstrate that handling ED problems with the proposed framework is a promising research direction.

Categories and Subject Descriptors

G.1.6 [Numerical Analysis]: Optimization—*Global optimization, constrained optimization*; J.0 [Computer Applications]: General

General Terms

Algorithms, Management, Performance

Keywords

Economic dispatch, Valve-point effect, Genetic algorithm, ECGA, Adaptive discretization, Split-on-demand, SoD

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '07, July 7–11, 2007, London, England, United Kingdom.
Copyright 2007 ACM 978-1-59593-697-4/07/0007 ...\$5.00.

1. INTRODUCTION

As the energy crisis is coming in the foreseeable future, especially the shortage of oil and other natural resources, studies on power systems become more and more important. Among all aspects of power systems, including generation, reservation, distribution, transmission, etc., the economic dispatch (ED) problem, consisting of several different aspects, is an important function in the power system operation. Economic dispatch refers to the problem of appropriately allocating generation among the generators to satisfy the specified constraints, such as the minimum output and maximum output of each generator, as well as to meet the exact given power demand. Traditional mathematical approaches, such as Lagrangian multipliers, cannot be used to solve the ED problem for modern generation units because the cost function for modern generation units does not possess the property required by the traditional approaches, such as monotonic increase. Moreover, the cost function of the ED problem is usually highly nonlinear and has a lot of local optima. As a consequence, even if there exists some way to transform the cost function to certain form that can be handled by a traditional method, the obtained solutions may still be the local optima instead of the global one due to the approximation or relaxation.

Thanks to the importance of and the challenge posed by the ED problem, researchers have been making numerous attempts to find or to develop suitable methodologies for obtaining better solutions. Among the promising sets of optimization techniques for tackling the ED problem are the methods proposed in the domain of evolutionary computation (EC). According to the optimization paradigm of most EC methods, the cost function of an ED problem can be simply considered as a black-box to optimize, and therefore, no special requirement for the cost function to satisfy. There have been a number of studies on the use of EC methods to handle the ED problem, including evolutionary programming (EP) [15, 9, 12], genetic algorithms (GAs) [14, 11, 4, 13, 2], and particle swarm optimization (PSO) [8], and the like. In this research project, we continue to explore the possibilities along the line and successfully employ a new integrated framework in the domain of evolutionary computation. In particular, the framework consists of an optimization engine: the extended compact genetic algorithm (ECGA), designed for binary variables, and an interface: split-on-demand (SoD), discretizing the continuous variables. The integration of ECGA and SoD solves a 3-unit ED problem as other algorithms do and obtains the currently best known solution for a 40-unit ED problem.

The remainder of the paper is organized as follows. Section 2 presents the formulation for the ED problem adopted in this study. Section 3 describes in detail the real-coded ECGA, which is composed of ECGA and SoD, as well as the constraint handling method. The ED problems for experiments introduced and the numerical results are shown in section 4, followed by the summary and conclusions of this paper in section 5.

2. ECONOMIC DISPATCH

The problem of economic dispatch (ED) for power systems is to find the optimal combination of power generations that minimizes the total generation cost while satisfying the specified equality and inequality constraints. In order to model the ED problem, a simplified cost function [1] of each generator which is represented as a quadratic function can be described as

$$C = \sum_{j \in J} F_j(P_j),$$

$$F_j(P_j) = a_j P_j^2 + b_j P_j + c_j, \quad (1)$$

where

- C : the total generation cost;
- J : the set for all generators;
- P_j : the electrical output of generator j ;
- F_j : the cost function for generator j ;
- a_j, b_j, c_j : the cost coefficients for generator j .

In the real world, the total generation have to be equal to the total system power demand plus the transmission network loss. However, for simplicity, the network loss is not considered in this work as in many studies. Thus, the constraints for the ED problem include two main parts. The first part is the equality constraint. The total system demand must be equal to the sum of the output of all generators, which can be formulated as

$$D = \sum_{j \in J} P_j, \quad (2)$$

where D is the total system demand.

The other part is the inequality constraint. The generation output of each generator has to be in the range of its minimum output and maximum output. An the inequality constraint for generator j can be put as

$$P_{jmin} \leq P_j \leq P_{jmax}, \quad (3)$$

where P_{jmin} and P_{jmax} are the minimum and maximum output of generator j , and P_j is the desired output.

In reality, the objective function of the ED problem is more complicated because of the valve-point effects, the change of fuels, and the other practical factors. Therefore, the nonsmooth cost functions should be considered instead of Equation (1), which is the most simplified form. The inclusion of the valve-point loading effects makes the modeling of the incremental fuel cost function of the generators more practical and closer to that in the real world. Such a model modification increases the non-linearity as well as the number of local optima in the solution space and makes the search algorithm easily trapped in the local optima. The

incremental fuel cost function of the generating units with the valve-point loadings [14] can be represented as

$$F_j(P_j) = a_j P_j^2 + b_j P_j + c_j + |e_j \sin(f_j \times (P_{jmin} - P_j))|, \quad (4)$$

where e_j and f_j are the coefficients for generator j to reflect the valve-point effects. In this paper, we focus on solving the ED problem with the valve-point effects, which is modeled as Equation (4).

3. REAL-CODED ECGA FOR ED

In this section, we will describe in detail the *real-coded extended compact genetic algorithm* (rECGA), which is an integration framework of an optimization engine—the *extended compact genetic algorithm* (ECGA) [5]—and an adaptive discretization method—*Split-on-Demand* (SoD) [3]. We will first introduce ECGA and SoD, followed by the integration of the two major components. Then, we will present the constraint handling technique employed in the study for handling the equality constraint of the ED problem.

3.1 ECGA for Optimization

The extended compact genetic algorithm (ECGA), proposed by Harik [5], is based on the idea that probability distributions can be used to model the population in genetic algorithms and the choice of a good probability distribution is equivalent to learning linkage between decision variables. The probabilistic models adopted in ECGA are a class of probabilistic models known as the marginal product models (MPMs). ECGA uses MPMs to model partitions of decision variables. The measurement of distribution quality is quantified based on the minimum description length (MDL) principle [10], which can be considered as a realization of *Occam's razor*. The key concept of MDL is that all things being equal, simpler distributions are better than more complex ones. The MDL restriction penalizes both inaccuracy and complexity, thereby leading to high quality probability distributions.

ECGA can be algorithmically outlined as

1. Initialize a population of size N at random.
2. Apply tournament selection of size S .
3. Model the population by using a greedy MPM search.
4. Stop if the MPM model has converged.
5. Generate a new population with the MPM model.
6. Return to step 2.

The complexity measurement of MPM is the sum of Model Complexity, formulated as Equation (5), and Compressed Population Complexity, formulated as Equation (6).

$$\text{Model Complexity} = \log N \sum_I 2^{S[I]}, \quad (5)$$

where N is the population size, and $S[I]$ is the length of the I th subset of genes.

$$\text{Compressed Population Complexity} = N \sum E(M_I), \quad (6)$$

where $E(M_I)$ is the entropy of the marginal distribution of subset I . According to the MDL principle, the goal for the greedy MPM search is to find an MPM model which has the minimal combined complexity:

$$\text{Combined Complexity} = \text{Model Complexity} + \text{Compressed Population Complexity} .$$

Instead of using traditional crossover and mutation operators, ECGA generates the offspring population from the MPM obtained in step 3. By doing such an operation, new individuals are generated without breaking building blocks represented in the form of gene groups. In ECGA, the original framework can only handle binary decision variables. In order to handle the ED problem, of which the decision variables are real numbers, certain technique is in order to interface the optimization engine with the problem. In this study, for the optimization functionality, we employ an ECGA extension, iECGA [6], proposed to directly handle integer variables in ECGA. For the variable-type interface, we adopt an adaptive discretization method, call split-on-demand, which is described in the next section.

3.2 Split on Demand for Discretization

Split-on-demand (SoD), proposed by Chen et al. [3], is an adaptive discretization method that splits each dimension of a real number into several intervals and encodes these intervals with integers. The principle of SoD is to split the real-number interval in which there are more than a certain number of search points. In order to determine which real-number interval to split, a *split rate* γ , where $0 < \gamma < 1$, is employed. Let the population size be N . If an interval contains more than $N \times \gamma$ search points, the interval should be split. By adjusting the split rate, we can control the accuracy of the probabilistic model that we want to build to describe the population as well as avoid using unnecessarily long bit-strings for real-number discretization.

When all split operations are done, i.e., no interval contains more than $N \times \gamma$ individuals, we decrease the split rate by a factor, ϵ , where $0 < \epsilon < 1$. Having a high split rate is similar to roughly dissecting the search space, while having a low split rate is similar to dissecting the search space in detail. Hence, the reason to manipulate the split rate in such a manner is that we would like to keep the population diversity and conduct a coarse-grained global search at the early search stage, to obtain more and more information regarding the solution space and know where to put the search points for finding good solutions as the search process goes by, and at the late search stage, to build accurate probabilistic models for conducting a fine-grained local search. The factor ϵ can be used to control the speed of convergence. An appropriate ϵ can help the search algorithm to avoid wasting time on useless regions as well as being trapped at local optima and therefore is key to an successful, efficient search process. The pseudo code for Split-on-Demand is shown Figure 1.

3.3 Real-Coded ECGA = ECGA with SoD

The basic idea of the proposed framework for real-coded ECGA (rECGA) is to employ SoD to discretize the population consisting of individuals of continuous decision variables and transform the individuals to those of integer variables. Then, we use ECGA to build marginal product models (MPMs) for the transformed individuals and generate the next generation by sampling the constructed MPMs. The

```

1: procedure SPLIT-ON-DEMAND
2:   Split(lower_bound, upper_bound)
3:    $\gamma \leftarrow \gamma \times \epsilon$ 
4: end procedure

1: procedure SPLIT( $\ell$ ,  $u$ )
2:    $m \leftarrow \text{random}[\ell, u]$ 
3:    $N_\ell \leftarrow$  number of individuals in  $[\ell, m]$ 
4:    $N_u \leftarrow$  number of individuals in  $[m, u]$ 
5:   if  $N_\ell \geq N \times \gamma$  then
6:     Split( $\ell$ ,  $m$ )
7:   else
8:     Add a code for the range  $[\ell, m]$ 
9:   end if
10:  if  $N_u \geq N \times \gamma$  then
11:    Split( $m$ ,  $u$ )
12:  else
13:    Add a code for the range  $[m, u]$ 
14:  end if
15: end procedure

```

Figure 1: Pseudo code for SoD.

procedure of rECGA can be put as the following steps:

1. Initialize a population of size N at random.
2. Apply tournament selection of size S .
3. Use SoD to encode each dimension of the variables.
4. Model the population composed of the encoded individuals by using a greedy MPM search.
5. Stop if the MPM model has converged.
6. Generate a new population with the MPM model.
7. Return to step 2.

In rECGA, we use SoD to encode each dimension of the individuals in the population after tournament selection and do the MPM greedy search as in ECGA. At the end of each iteration, local search operators may be employed to improve the obtained solutions with a probability, but in the present work, no local search operator is adopted such that we can more accurately assess the optimization capability of the combination of ECGA and SoD. Furthermore, some constraint handling techniques have to be utilized to handle the equality constraint in the ED problem, and those adopted in the study will be described in the next section.

3.4 Constraint Handling

One of the most important topic for solving ED problems is the equality and inequality constraints. These constraints divide the entire solution space into lots of complicated regions. Such a situation prevents many search techniques and optimization algorithms from performing effectively and efficiently on ED problems. Hence, in this study, we devise a constraint handling technique specifically for the ED problem based on the concept of repair. Repairing solutions means transforming infeasible solutions into feasible ones in some way. For the ED problem, although the inequality constraints (Equation (3)) might need handling in traditional mathematical programming methods, they can

Generator	P_{min} (MW)	P_{max} (MW)	a	b	c	e	f
1	100	600	0.001562	7.92	561	300	0.0315
2	100	400	0.00482	7.97	78	150	0.063
3	50	200	0.00194	7.85	310	200	0.042

Table 1: Parameters for test case I (3-unit system) with the valve-point loading effect. a , b , c , e , and f are the cost coefficients in the fuel cost function: $F_j(P_j) = a_j P_j^2 + b_j P_j + c_j + |e_j \sin(f_j \times (P_{jmin} - P_j))|$.

be thoroughly ignored in the framework because the control of decision variable ranges is a built-in functionality of EC methods. As for the equality constraint (Equation (2)), we repair infeasible solutions in the following procedure. Firstly, we generate a number sequence from 1 to the number of generator with a uniformly distributed random order. Each number in the sequence denotes a generator in the solution which needs repair. The whole sequence means the order in which we process the specified generator. As an example, for five generators, if we randomly generate a sequence: 3, 2, 5, 1, 4, the sequence means that we will firstly process unit 3, then unit 2, unit 5, and so on. In the specified order, we check the equality constraint, i.e., the sum of the total power output has to be equal to the system demand. If the equality constraint is not satisfied, the output of the current generator is modified according to

$$P'_i = \min(\text{UBound}(P_i), \max((D - \sum_{j=1, j \neq i}^n P_j), \text{LBound}(P_i))),$$

where D is the system power demand, $\text{LBound}(P_i)$ and $\text{UBound}(P_i)$ are the lower bound and upper bound of P_i , i.e., the inequality constraint of P_i .

The proposed framework, ECGA with SoD, incorporating the constraint handling technique is able to solve the ED problem effectively. After adopting the described repair mechanism, the real-coded ECGA for tackling the ED problem can be outlined as the following steps:

1. Initialize a population of size N at random according to the constraints posed to the generator output.
2. Apply tournament selection of size S .
3. Use SoD to encode each dimension of the variables.
4. Model the population composed of the encoded individuals by using a greedy MPM search.
5. Stop if the MPM model has converged.
6. Generate a new population with the MPM model.
7. Repair the infeasible individuals in the population.
8. Return to step 2.

In order to observe the effectiveness and to verify the performance of the proposed approach, two ED problems, one consisting of 3 generators and the other consisting of 40 generators, are served as a testbed. The experimental results on the two ED problems are presented in the next section.

4. EXPERIMENTS AND RESULTS

In this study, we focus on solving the ED problem with nonsmooth cost functions considering the valve-point effects. The nonsmooth cost functions were described in section 2 as Equation (4). To examine the performance, the real-coded ECGA for ED problems proposed in section 3.4 is applied to two ED problems which were adopted as test problems in the literature [14, 12]. One consists of 3 generator units, and the other consists of 40 generator units. The input data for the 3-generator system are given by Walters and Sheble [14], and those for the 40-generator system are given by Sinha et al. [12], respectively. The detailed parameters for the two test problems, including the lower bound and upper bound for the output of each generator as well as the coefficients for computing the cost functions, are given in Tables 1 and 2. The total power demand for the 3-unit system is 850MW, and the demand for the 40-unit system is 10500MW. It has been proven that for the 3-unit system given by Table 1, the global optimal solution is 8234.07 [7]. As for the 40-unit system, the global optimal solution has not been determined. To our limited knowledge, the known best solution reported in the literature is 122252.265 [8].

To conduct the experiments, the parameter settings for the real-coded ECGA are that population size = 400, probability of crossover = 0.975, tournament size = 8, $\gamma = 0.5$, $\epsilon = 0.999$, and the maximum fitness evaluations is 200000. For each problem, 100 independent trails were conducted to collect statistically significant results. The obtained experimental results for the 3-unit system are given in Table 3 and are compared to those obtained by IEP [9], EP [15], and MPSO [8]. The results for the small ED problem demonstrate that ECGA with SoD was able to find the global optimal solution presented by Lin et al. [7].

For the 40-unit system, the experimental results are compared with those obtained by other methods described by Sinha et al. [12], such as classical EP (CEP), fast EP (FEP), modified FEP (MEFP), improved FEP (IFEP), as well as the results obtained by MPSO [8]. The minimum costs, i.e., the best solutions, achieved by each method are shown in Table 4. The best solution obtained by the real-coded ECGA is 121462.3591, which is better than the known best result, 122252.265, reported in the literature [8]. For access and verification purpose, the generator outputs and the corresponding costs of the best solution obtained by rECGA are provided in Table 5.

Because of the stochastic nature of the methods in evolutionary computation, in order to avoid reporting the result of a ‘‘lucky shot’’ in the 100 independent trials, comparison of the experimental results in a statistical manner should be implemented. First of all, Table 6 shows the range of the results in 100 trials obtained by CEP, FEP, MFEP, IFEP, MPSO, and rECGA, where the listed results except for that of rECGA are provided in [12, 8]. As we can observe in Ta-

Generator	P_{min} (MW)	P_{max} (MW)	a	b	c	e	f
1	36	114	0.0069	6.73	94.705	100	0.084
2	36	114	0.0069	6.73	94.705	100	0.084
3	60	120	0.2028	7.07	309.54	100	0.084
4	80	190	0.00942	8.18	369.03	150	0.063
5	47	97	0.0114	5.35	148.89	120	0.077
6	68	140	0.01142	8.05	222.33	100	0.084
7	110	300	0.00357	8.03	287.71	200	0.042
8	135	300	0.00492	6.99	391.98	200	0.042
9	135	300	0.00573	6.6	455.76	200	0.042
10	130	300	0.00605	12.9	722.82	200	0.042
11	94	375	0.00515	12.9	635.2	200	0.042
12	94	375	0.00569	12.8	654.69	200	0.042
13	125	500	0.00421	12.5	913.4	300	0.035
14	125	500	0.00752	8.84	1760.4	300	0.035
15	125	500	0.00708	9.15	1728.3	300	0.035
16	125	500	0.00708	9.15	1728.3	300	0.035
17	220	500	0.00313	7.97	647.85	300	0.035
18	220	500	0.00313	7.95	649.69	300	0.035
19	242	550	0.00313	7.97	647.83	300	0.035
20	242	550	0.00313	7.97	647.81	300	0.035
21	254	550	0.00298	6.63	785.96	300	0.035
22	254	550	0.00298	6.63	785.96	300	0.035
23	254	550	0.00284	6.66	794.53	300	0.035
24	254	550	0.00284	6.66	794.53	300	0.035
25	254	550	0.00277	7.1	801.32	300	0.035
26	254	550	0.00277	7.1	801.32	300	0.035
27	10	150	0.52124	3.33	1055.1	120	0.077
28	10	150	0.52124	3.33	1055.1	120	0.077
29	10	150	0.52124	3.33	1055.1	120	0.077
30	47	97	0.0114	5.35	148.89	120	0.077
31	60	190	0.0016	6.43	222.92	150	0.063
32	60	190	0.0016	6.43	222.92	150	0.063
33	60	190	0.0016	6.43	222.92	150	0.063
34	90	200	0.0001	8.95	107.87	200	0.042
35	90	200	0.0001	8.62	116.58	200	0.042
36	90	200	0.0001	8.62	116.58	200	0.042
37	25	110	0.0161	5.88	307.45	80	0.098
38	25	110	0.0161	5.88	307.45	80	0.098
39	25	110	0.0161	5.88	307.45	80	0.098
40	242	550	0.00313	7.97	647.83	300	0.035

Table 2: Parameters for test case II (40-unit system) with the valve-point loading effect. a , b , c , e , and f are the cost coefficients in the fuel cost function: $F_j(P_j) = a_j P_j^2 + b_j P_j + c_j + |e_j \sin(f_j \times (P_{jmin} - P_j))|$.

Generator	GA	IEP (pop=20)	EP	MPSO (par=20)	rECGA
1	300	300.23	300.26	300.27	300.267
2	400	400	400	400	400
3	150	149.77	149.74	149.73	149.733
TP	850	850	850	850	850
TC	8237.6	8234.09	8234.07	8234.07	8234.07

Table 3: Comparison of the experimental results obtained by various methods on the nonsmooth cost function considering the valve-point loading effect. For the 3-unit system, IEP, EP, MPSO, and rECGA were able to find the global optimum [7].

	CEP	FEP	MFEP	IFEP	MPSO	rECGA
Minimum Cost	123488.3	122679.7	122647.6	122624.35	122252.265	121462.3591

Table 4: Comparison of the experimental results obtained by various methods on the nonsmooth cost function considering the valve-point loading effect. For the 40-unit system, rECGA was able to find the best solution.

Generator	P_{min} (MW)	P_{max} (MW)	Output	Cost
1	36	114	110.80098	925.11565
2	36	114	110.88806	926.56631
3	60	120	97.40449	1190.63739
4	80	190	179.73300	2143.55011
5	47	97	96.15215	840.66343
6	68	140	140.00000	1596.46432
7	110	300	299.99898	3216.41474
8	135	300	284.62219	2780.24662
9	135	300	284.61234	2798.46198
10	130	300	130.00001	2502.06532
11	94	375	94.00003	1893.30606
12	94	375	94.00027	1908.17291
13	125	500	214.76169	3792.11715
14	125	500	394.27878	6414.85790
15	125	500	304.52026	5171.21428
16	125	500	394.28449	6436.71537
17	220	500	489.27966	5296.71703
18	220	500	489.27855	5288.76474
19	242	550	511.27996	5540.94200
20	242	550	511.28163	5540.95823
21	254	550	523.28030	5071.30855
22	254	550	523.28419	5071.38735
23	254	550	523.28495	5057.33548
24	254	550	523.28151	5057.26621
25	254	550	523.28214	5275.14526
26	254	550	523.27977	5275.09678
27	10	150	10.00013	1140.52698
28	10	150	10.00517	1140.64280
29	10	150	10.00018	1140.52812
30	47	97	87.84287	707.21302
31	60	190	189.99927	1643.98840
32	60	190	189.99996	1643.99109
33	60	190	189.99993	1643.99098
34	90	200	199.99994	2101.01644
35	90	200	199.99993	2043.72638
36	90	200	199.99972	2043.72436
37	25	110	110.00000	1220.16612
38	25	110	109.99978	1220.16484
39	25	110	109.99871	1220.15859
40	242	550	511.28401	5541.02984
Total Generation & Total Cost			10500	121462.3591

Table 5: The generator outputs and the corresponding costs of the best solution obtained by rECGA.

Method	Range of Cost											
	127.0	126.5	126.0	125.5	125.0	124.5	124.0	123.5	123.0	122.5	122.0	121.5
	-	-	-	-	-	-	-	-	-	-	-	-
	126.5	126.0	125.5	125.0	124.5	124.0	123.5	123.0	122.5	122.0	121.5	121.0
CEP	10	4	-	16	22	42	4	2	-	-	-	-
FEP	6	-	4	2	10	20	26	24	6	-	-	-
MFEP	-	-	-	-	-	14	26	50	10	-	-	-
IFEP	-	-	2	-	4	4	18	50	22	-	-	-
MPSO	-	-	-	-	-	-	-	-	53	47	-	-
rECGA	-	-	-	-	-	-	-	-	-	2	97	1

Table 6: Comparison of method on relative frequency of convergence in the ranges of cost

	rECGA	MPSO
mean	121777.649963	122516.06455
<i>t</i> -value	27.8068829451749	
<i>p</i> -value	2.2645299161711E-55	

Table 7: The t-test for the experimental results obtained by rECGA and MPSO under condition 1, where the rECGA data set contains the actual results, and the MPSO data set contains forty-seven 122252.265 and fifty-three 122750.

ble 6, the distribution of the results can be considered better than those for other evolutionary algorithms. Furthermore, to precisely compare the performance of rECGA and MPSO [8] on the 40-unit ED problem, the t-test was conducted to indicate the statistical significance of the obtained results. Since the actual data set of the 100 trials for MPSO is not available, in order to get a fair comparison and assessment, we set up two conditions under which the t-test can be conducted. According to the data given in Table 6, the first condition is that the MPSO data set contains forty-seven 122252.265, which is the optimum reported for MPSO [8], and fifty-three 122750, which is the mean value of 122500 and 123000. Table 7 demonstrates the t-test results for condition 1. Given the *p*-value: 2.26×10^{-55} , which is much smaller than the commonly used significant levels, such as 0.05 (5%), 0.01 (1%), or 0.001 (0.1%), we can conclude that the performance of rECGA on the 40-unit ED problem is statistically significantly better than that of MPSO on the same problem. For condition 2, the MPSO data set contains forty-seven 122252.265, which is the optimum reported for MPSO [8], and fifty-three 122500, which is the best value in the range from 122500 to 123000. The t-test results under condition 2 are shown in Table 8. Due to the change of the standard deviation, the *p*-value becomes 9.09×10^{-91} . Such a small *p*-value prevents us from accepting the null hypothesis, which in this case is interpreted as that the performance of rECGA and MPSO on the test problem is equivalent.

According to the results, we can know that the proposed algorithm performed well on the two test ED problems. In particular, for the 40-unit system, we improved the known best solution from 122252.265 [8] to 121462.3591. The real-coded ECGA is capable of solving ED problems effectively.

5. SUMMARY AND CONCLUSIONS

In this work, we employed the extended compact genetic algorithm (ECGA) as an optimization engine and split-on-

	rECGA	MPSO
mean	121777.649963	122383.56455
<i>t</i> -value	39.4214198098397	
<i>p</i> -value	9.0857670116394E-91	

Table 8: The t-test for the experimental results obtained by rECGA and MPSO under condition 2, where the rECGA data set contains the actual results, and the MPSO data set contains forty-seven 122252.265 and fifty-three 122500.

demand (SoD), which is an adaptive discretization method, as a variable-type interface. By combing ECGA and SoD, we proposed the framework of the real-coded ECGA (rECGA) and used rECGA on two economic dispatch (ED) problems for examining the performance. Incorporating with the proposed constraint handling technique, rECGA successfully achieved the global optimal solution of the ED problem consisting of 3 generators and was able to obtain the solutions better than the known best solution reported in the literature for the 40-unit ED problem. Furthermore, the t-test was conducted to demonstrate that the performance of rECGA is statistically significant.

The overall results of this study serves two important purposes. The first one is that ECGA with SoD is capable of appropriately handling the ED problem of which the cost function is nonsmooth with the valve-point effect. Since the energy crisis comes closer and closer, power related problems, such as the ED problem, become more and more important. Although the ED problems considered in this study are not the most complicated ones, it is still a promising research direction to employ the proposed framework to tackle the similar problems such that some parts of current power systems might be improved or enhanced.

On the other hand, the results of this study also demonstrate that it is viable to employ an optimization algorithm designed for handling decision variables of the discrete type to handle problems consisting of continuous variables, as long as a suitable interface is adopted. Although many researchers do not consider the variable-type transformation as an issue, in practice, except for some limited cases, most algorithms for discrete variables do not perform well on continuous problems and vice versa. By comparing the real-coded ECGA to the algorithms designed for continuous variables, such as particle swarm optimization (MPSO) and evolutionary programming (IFEP, MFEP, FEP, CEP), this paper provides solid experimental results to serve as the

proof of principle for transforming the variable type while retaining the capability of the optimization algorithm.

Finally, the future work includes applying the proposed framework to other important problems as well as developing different integrations of optimization engines and data type transforming techniques. Theoretical understandings for the quality of the transforming techniques, such as SoD and fixed-height histogram, as well as for the interaction between the engine and the interface can also be considered.

Acknowledgments

The work was partially sponsored by the National Science Council of Taiwan under grants NSC-95-2221-E-009-092 and NSC-95-2627-B-009-001 as well as by the MOE ATU Program. The authors are grateful to the National Center for High-performance Computing for computer time and facilities.

6. REFERENCES

- [1] J. W. Allen and F. W. Bruce. *Power Generation, Operation, and Control*. New York: Wiley, 1984.
- [2] A. G. Bakirtzis, P. N. Biskas, C. E. Zoumas, and V. Petridis. Optimal power flow by enhance genetic algorithm. *IEEE Transactions on Power Systems*, 17:229–236, May 2002.
- [3] C.-H. Chen, W.-N. Liu, and Y.-p. Chen. Adaptive discretization for probabilistic model building genetic algorithms. In *Proceedings of ACM SIGEVO Genetic and Evolutionary Computation Conference (GECCO-2006)*, pages 1103–1110, 2006.
- [4] P.-H. Chen and H.-C. Chang. Large-scale economic dispatch by genetic algorithm. *IEEE Transactions on Power Systems*, 10:1919–1926, February 1995.
- [5] G. R. Harik. Linkage learning via probabilistic modeling in the ECGA. IlliGAL Report No. 99010, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL, 1999.
- [6] P.-C. Hung and Y.-p. Chen. iECGA: Integer extended compact genetic algorithm. In *Proceedings of ACM SIGEVO Genetic and Evolutionary Computation Conference (GECCO-2006)*, pages 1415–1416, 2006.
- [7] W. M. Lin, F. S. Cheng, and M. T. Tsay. An improved tabu search for economic dispatch with multiple minima. *IEEE Transactions on Power Systems*, 17(1):108–112, 2002.
- [8] J. B. Park, K. S. Lee, J. R. Shin, and K. Y. Lee. A particle swarm optimization for economic dispatch with nonsmooth cost functions. *IEEE Transactions on Power Systems*, 20(1):34–42, 2005.
- [9] Y.-M. Park, J. R. Won, and J. B. Park. New approach to economic load dispatch based on improved evolutionary programming. *Eng. Intell. Syst. Elect. Eng. Commun*, 6(2):103–110, June 1998.
- [10] J. Rissanen. *Stochastic Complexity in Statistical Inquiry*. World Science, 1989.
- [11] G. B. Sheble and K. Brittig. Refined genetic algorithm-economic dispatch example. *IEEE Transactions on Power Systems*, 10:117–124, November 1995.
- [12] N. Sinha, R. Chakrabarti, and P. K. Chattopadhyay. Evolutionary programming techniques for economic load dispatch. *IEEE Transactions on Evolutionary Computation*, 7(1):83–94, February 2003.
- [13] K. S. Swarup and S. Yamashiro. Unit commitment solution methodologies using genetic algorithm. *IEEE Transactions on Power Systems*, 17:87–91, February 2002.
- [14] D. C. Walters and G. B. Sheble. Genetic algorithm solution of economic dispatch with valve point loading. *IEEE Transactions on Power Systems*, 8(3):1325–1332, August 1993.
- [15] H. T. Yang, P. C. Yang, and C. L. Huang. Evolutionary programming based economic dispatch for units with non-smooth fuel cost functions. *IEEE Transactions on Power Systems*, 11(1):112–117, 1996.