A DIFFERENTIAL EVOLUTION APPROACH FOR THE UNIT COMMITMENT PROBLEM

Ali Keleş [1] — *kelesal@itu.edu.tr*

A. Şima Etaner-Uyar [1] — *etaner@itu.edu.tr*

Belgin Türkay [2] — *turkay@elk.itu.edu.tr*

[1] *Istanbul Technical University, Department of Computer Engineering, 34490, Maslak, Istanbul, Turkey*
[2] *Istanbul Technical University, Department of Electrical Engineering, 34490, Maslak, Istanbul, Turkey*

Key words: Differential evolution, binary differential evolution, unit commitment problem

ABSTRACT

This paper proposes a Differential Evolution (DE) approach for the Unit Commitment Problem (UCP). The proposed approach is tested on benchmark UCP datasets as well as on real-world data obtained from the Turkish interconnected power network system. The results of the DE on the benchmark datasets are comparable with the results of a current state-of-the-art evolutionary approach found in literature. This preliminary experimental study shows that DE is suitable for the UCP and the promising results promote further study.

I. INTRODUCTION

The Unit Commitment Problem (UCP) is a constrained optimization problem in which, optimal turn-on and turn-off schedules need to be determined over a given time horizon for a group of power generation units under some operational constraints. The objective is to minimize the power generation costs while meeting the hourly forecasted power demands. The UCP is an important area of research which has attracted increasing interest from the scientific community due to the fact that even small savings in the operation costs for each hour can lead to major overall economic savings.

The UCP consists of two sub-problems [15]: In the first part, a feasible, low-cost schedule for turn-on and turn-off times of the power generation units over the given time horizon is determined. In the second part, for each hour, the power outputs for the units scheduled to be online for that hour are obtained in such a way as to minimize the fuel costs while meeting the forecasted power demands for that hour. This second part is termed as the Economic Dispatch Problem (14). Several approaches exist in literature to tackle the UCP, such as dynamic-programming [1, 2], Lagrangian relaxation [3], branch and bound [4], benders decomposition [5], simulated annealing [6], tabu-search [7], evolutionary algorithms [8, 9, 10, 11, 15] and many hybrids. A detailed survey can be found in [16].

Evolutionary Algorithms (EAs) [19] are population based optimization techniques based on mechanisms found in nature. The Differential Evolution (DE) [13] algorithm, introduced by Storn and Price in 1995, belongs to the group of evolutionary algorithms which operate in continuous search spaces. DE has been successfully applied to many problem domains. The solution to the UCP is given as a set of binary decision variable assignments showing which generator units are online and which are offline for any given time slot. This makes it impossible to apply a pure DE to the UCP. Therefore, in this study a binary version of DE (BDE) is used to solve the schedule determination part of the UCP. For the EDP, a standard lambda-iteration method [12] is used. BDE is tested on benchmark UCP data as well as on real-world data of the Turkish interconnected power system.

This paper is organized as follows: In section 2, the UCP is explained. Section 3 introduces the BDE approach used in this study. In section 4, experiments and results are given. Section 5 concludes the paper.

II. THE UNIT COMMITMENT PROBLEM

The objective of the UCP is to minimize the total cost of power generation over a given time horizon. Three main factors effect this cost: Fuel costs, start-up costs and operational constraints of units. The parameters used in the UCP formulation are as follows:

$P_i(t)$: generated power by unit i at time t
$F_i(p)$: cost of producing p MW power by unit i
PD(t): power demand at time t
PR(t): power reserve at time t
$CS_i(t)$: start-up cost of *i*-th unit at time t

$x_i(t)$: duration that unit i has stayed offline since hour t
$v_i(t)$: status of *i*-th unit at time t (online-offline)

Fuel cost depends on the amount of power output provided by each online unit for each time slot. The fuel cost needs to be minimized subject to two constraints: The power demands for each hour should be met and the power generated by each unit should be within its minimum and maximum capacities. This part of the objective can be formulized as follows.

$$\min \quad F_{total}(t) = \sum_{i=1}^{N} F_i(P_i(t))$$

Subject to constraints:

$$\sum_{i=1}^{N} P_i(t) = PD(t)$$

$$P_i^{\min} \le P_i(t) \le P_i^{\max}$$

Start-up costs depend upon the number of hours a unit has been down. The formulation for the start-up cost is:

$$CS_i(t) = \begin{cases} CS_{hot} & if \quad x_i(t) \le t_{coldstart} \\ CS_{cold} & otherwise \end{cases}$$

There is also another constraint named *minimum up/down*. Each generator should stay online for an arbitrary number of hours after it is turned-on. It also should stay off for a time after it is turned-off. This number of hours can vary according to different power generating units. The formulation for these constraints is:

$$\begin{array}{ll} if\ v_i(t) = 1 & x_i(t-1) \ge t_{down} \\ else & x_i(t-1) \ge t_{up} \end{array}$$

According to these fuel cost and start-up cost functions and constraints, the formulation for the UCP for N units and T hours is as given below:

$$\min \quad F_{total} = \sum_{t=1}^{T} \sum_{i=1}^{N} \left[F_i(P_i(t)).v_i(t) + CS_i(t) \right]$$

Subject to constraints:

$$\sum_{i=1}^{N} P_i(t).v_i(t) = PD(t)$$

$$v_i(t).P_i^{\min} \le P_i(t) \le v_i(t) P_i^{\max}$$

$$\sum_{i=1}^{N} P_i^{\max}(t).v_i(t) \ge PD(t) + PR(t)$$

$$\begin{array}{ll} if\ v_i(t) = 1 & x_i(t-1) \ge t_{down} \\ else & x_i(t-1) \ge t_{up} \end{array}$$

The fuel cost of generating *p* MW power for the *i*-th unit is calculated using the following formula:

$$F_i(p) = a_{0i} + a_{1i}.p + a_{2i}.p^2$$

As can be seen, this cost for a generating unit depends on three parameters: a_{0i}, a_{1i} and a_{2i}. The lambda-iteration technique [12, 14] uses this formulation to find the lowest cost for dispatching the amount of power to be generated by the online generating units. This corresponds to the EDP. To solve the EDP by lambda-iteration, an optimal lambda value which also satisfies the constraints is searched for.

III. BINARY DIFFERENTIAL EVOLUTION

The Differential Evolution (DE) [13] algorithm was introduced by Storn and Price in 1995. DE is a form of an evolutionary algorithm which operates in continuous search spaces. DE is based on four main steps: Initialization, mutation, recombination and selection. While the initialization step is only done in the first iteration, the other three steps take place in each iteration. All individuals pass through these operations.

The chromosomes of an individual are made up of real valued genes, $X_{j,i,g}$ (where j is the index of the parameter, i is the index of the individual and g shows the generation number), each of which correspond to the parameters of the problem to be optimized.

All individuals in the population, called the *target vectors*, go through the mutation and recombination steps. There are several mutation operators. One of the most commonly used forms of these operators, the DE/rand/1 method, chooses three different vectors from the population and creates a mutant vector from these, called the *donor vector*, through the equation given below.

$$V_{i,g} = X_{r0,g} + F(X_{r1,g} - X_{r2,g})$$

F takes values in the range (0,1+) and it is recommended to set F less than 1 [13]. As can be seen from the definition of the mutation operator of the DE algorithm, it is not possible to use it for binary valued problems without a modification. There are some approaches in literature for modifying DE for such binary valued problems. One of these methods uses an *angle modulation technique* to transform the binary space into a continuous space [17]. In the initial testing stage of this study, the experimental results obtained using this technique turned out to be insufficient. So the approach proposed in [18] is used as the BDE implementation in this study. The details of this algorithm are given below.

The initialization step randomly sets the initial values of the parameters in the population. to be either 1 or 0. The modification on DE to make it run within binary spaces, is done to the mutation operator. According to the approach proposed in [18], the multiplication, addition and substitution operators are changed as explained below. The value of any parameter in any of the vectors can be either 0 or 1. To preserve this property, the result of subtraction and the addition operators are obtained using the hamming distance between the two vectors. After the substitution step, each parameter in the vector is multiplied with the F parameter. This operation forces the values of the parameters to change from binary space to continuous space. In the next step of the mutation operator, which is the addition operator, the values are transformed back into being either 0 or 1 through a rounding mechanism. A sample application of the substitution operator is given in Table-1.

Table 1: Sample application of the substitution operator

Xr2,g	1	1	0	1	0
Xr1,g	0	1	0	0	0
After Substitution	1	0	0	1	0

The aim of the recombination operation is to create a different vector based on the donor and the target vectors. The parameters of this vector are taken from the target vector when a uniformly distributed random number is greater than a predefined *Cr* value; otherwise, it is taken from the donor vector [13] as shown in

$$U_{j,i,g} = \begin{cases} V_{j,i,g} & if\,(rand_j(0,1) \le Cr \;\; or \;\; j = j_{rand}) \\ X_{j,i,g} & otherwise \end{cases}$$

There are two proposed ways [18] to implement this step: binomial and exponential. The binomial crossover operation considers each parameter in a vector separately, however in the exponential crossover operation after ($rand_j(0,1) \le Cr$) becomes true for the first time, the remaining parameters are taken from the donor vector as a block. The binomial crossover operator is chosen in this study. Cr takes values in the range [0,1]. The vector that is created through the recombination step is called the *trial vector*.

In the selection step, either the target vector or the trial vector is chosen for the next generation as shown below.

$$X_{i,g+1} = \begin{cases} U_{i,g} & if\,(\,fitness(V_{i,g}) \le fitness(X_{i,g})\,) \\ X_{i,g} & otherwise \end{cases}$$

These steps continue until an acceptable solution is found or until a predefined number of maximum DE iterations has been reached.

IV. EXPERIMENTS

The results of the tests on the benchmark data will be compared to those of a state-of-the-art memetic algorithm proposed in [10] and a genetic algorithm. In the BDE, the population consists of 100 individuals, Cr is taken as 0.3 and F is taken as 0.8. In lambda iteration, the tolerance is set to 0.0001.

The fitness values of the individuals are calculated as the summation of the fuel cost, the cost of start-up and a penalty value. Cost for power generation is calculated using lambda-iteration based on the status of each power generator unit. For each hour, depending on whether the start-up is a cold start or a hot start, the appropriate cost is added to the total cost. A penalty term is used if the hourly power demands plus a specific amount of reserve is not met or if t_{up} and t_{down} constraints are violated. The multiplier M for the first penalty term is set to 200 for all tests. The multiplier K for the second penalty term is taken as 150 for the second and third tests but as 600 for the first test. Details on the fitness evaluation and the penalty calculation method can be found in [10]. Several runs of the algorithm are performed and the best results for the total cost are reported here. All parameter values are determined using the best settings found as a result of a series of experimental runs.

The first test problem [12] has four power generating units and a time horizon of eight hours. The data for this test system is given in Table-2 and Table-3. The results are presented in Table-4. The best overall result obtained using BDE is approximately 74,676 and the best result of the genetic algorithm is given as 74,675 in [10].

For the second test, a larger dataset [8] consisting of 10 generating units and a time horizon of 24 hours is used. The data and the results for this test are not reported here due to space restrictions. The best result of the genetic algorithm [10] is 565,866 and the best result of the memetic algorithm [10] is 565,827 as compared to the best BDE result which is 566,166 for this test set.

For the third test, real-world data from the Turkish interconnected network system is used. There are 8 generating units and a time horizon of 8 hours. The data for this test system is given in Table-5 and Table-6 and the results are given in Table-7.

An overview of the best total cost values for the three tests is summarized in Table-8. As can be seen, for the first test, which is smaller in problem size than the other two, BDE gives the same result as the genetic algorithm. For the second test, results of the memetic and the genetic algorithms are slightly better than the BDE. It should be kept in mind that while the BDE is in its most basic form, the memetic algorithm [10] uses extra local search through hill-climbing during its run which is costly. Thus BDE performs fewer actions to find comparable results to the state-of-the-art memetic algorithm.

Table 2: Test System 1 [12]

	Unit 1	Unit 2	Unit 3	Unit 4
P_{max}(MW)	300	250	80	60
P_{min}(MW)	75	60	25	20
a_0	684.74	585.62	213.0	252.0
a_1	16.83	16.95	20.74	23.60
a_2	0.0021	0.0042	0.0018	0.0034
t_{up}(h)	5	5	4	1
t_{down}(h)	4	3	2	1
S_{hot}($)	500	170	150	0
S_{cold}($)	1100	400	350	0.02
$t_{coldstart}$(h)	5	5	4	0
Initial State(h)	8	8	-5	-6

Table 3: Demand and Reserve for Test System 1 [12].

Hour	**1**	**2**	**3**	**4**
Demand	450	530	600	540
Reserve	45	53	60	54
Hour	**5**	**6**	**7**	**8**
Demand	400	280	290	500
Reserve	40	28	29	50

Table 4: Results for Test System 1.

	$P1_{out}$	$P2_{out}$	$P3_{out}$	$P4_{out}$
Hour 1	300.0	150.0	0	0
Hour 2	300.0	205.0	25.0	0.0
Hour 3	300.0	250.0	30.0	20.0
Hour 4	300.0	215.0	25.0	0.0
Hour 5	300.0	0.0	80.0	20.0
Hour 6	255	0.0	25.0	0.0
Hour 7	265.0	0.0	25.0	0.0
Hour 8	300.0	200.0	0.0	0.0

Table 5: Turkish Interconnected Power System Network

	U 1	U 2	U 3	U 4
P_{max}(MW)	1120	1350	1432	600
P_{min}(MW)	190	245	318	150
a_0	6595,5	7290,6	6780,5	1564,4
a_1	7,0063	7,2592	5,682	3,1288
a_2	0,0168	0,0127	0,0106	0,0139
t_{up}(h)	8	1	1	10
t_{down}(h)	2	0,5	0,5	3
S_{hot}($)	800	800	600	400
S_{cold}($)	1600	1600	1200	800
$t_{coldstart}$(h)	8	1	1	10
Initial State(h)	-4	-4	-4	-4
	U 5	**U 6**	**U 7**	**U 8**
P_{max}(MW)	990	420	630	630
P_{min}(MW)	210	110	140	140
a_0	5134,1	1159,5	1697	1822,8
a_1	6,232	3,3128	3,2324	3,472
a_2	0,0168	0,021	0,013	0,0147
t_{up}(h)	10	10	10	10
t_{down}(h)	3	3	3	3
S_{hot}($)	500	400	400	400
S_{cold}($)	1000	800	800	800
$t_{coldstart}$(h)	10	10	10	10
Initial State(h)	-4	-4	-4	-4

Table 6: Demand and Reserve for Turkish Interconnected Power System Network

Hour	**1**	**2**	**3**	**4**
Demand	2000	3000	6500	1500
Reserve	200	300	650	150
Hour	**5**	**6**	**7**	**8**
Demand	4200	5100	2700	1750
Reserve	420	510	270	175

Table 7: Results for Turkish Interconnected Power System Network

	$P1_{out}$	$P2_{out}$	$P3_{out}$	$P4_{out}$
Hour 1	0	0	0	549.4
Hour 2	0	0	740.7	600
Hour 3	844.5	1107	1400.9	600
Hour 4	190	0	0	304.6
Hour 5	511.7	0	873.5	600
Hour 6	559.2	729.8	948.8	600
Hour 7	339	0	0	549.2
Hour 8	191.5	0	0	371
	$P5_{out}$	**$P6_{out}$**	**$P7_{out}$**	**$P8_{out}$**
Hour 1	0	359.3	583.7	507.8
Hour 2	0	420	630	609.3
Hour 3	867.5	420	630	630
Hour 4	210	197.3	321.7	276.4
Hour 5	534.8	420	630	630

Hour 6	582.3	420	630	630
Hour 7	362	359	583.2	507.6
Hour 8	214.6	241.2	392.6	339

Table 8: Total Costs for Tests

	BDE	GA	Memetic
Test System 1	74,676	74,675	-
Test System 2	566,166	565,866	565,827
Turkish Interconnected Power System Network	532,142	-	-

V. CONCLUSION AND FUTURE WORK

The use of a binary differential evolution algorithm (BDE) for the unit commitment problem (UCP) is explored in this study. Three sets of tests are performed. The first two sets are on benchmark datasets obtained from the literature. The results of these two sets are compared to those of a current state-of-the-art evolutionary algorithm, namely a memetic algorithm, and a genetic algorithm found in literature. Then, the BDE is applied to the real-word data obtained from the Turkish interconnected network system. The results obtained for the benchmark tests are comparable to those of the memetic and genetic algorithms. As also stated above, it should be noted that while the BDE used in this study is in its most basic form, the memetic algorithm uses local search through hill climbing during its execution, which is costly. Thus BDE performs fewer actions to find comparable results. This preliminary experimental study shows that BDE is suitable for the UCP. Other, more complicated mechanisms like in the memetic algorithm, can be incorporated into the BDE to make it perform even better. Also, a sensitivity analysis for the parameter settings should be done. In this study, the best settings determined as a result of experimental runs is used, however it is seen that performance may be dependent on the selection of some of the parameters. This should be thoroughly explored. Also larger datasets can be used to test the scalability of BDE. Average performances of the different methods should also be compared in addition to the best performance. Overall, the BDE performs well on the UCP and the results promote further study.

REFERENCES

1. P. G. Lowery, "Generating Unit Commitment by Dynamic Programming", IEEE Transactions on Power Apparatus and Systems Vol. PAS-85, No. 5, pp. 422-426, 1966.
2. W. J. Hobbs, G. Hermon, S. Warner, G. B. Sheble, "An Enhanced Dynamic Programming Approach for Unit Commitment", IEEE Transactions on Power Systems, Vol 3, No. 3, pp. 1201-1205, 1988.
3. C. Cheng, C. W. Liu, and C. C. Liu, "Unit Commitment by Lagrangian Relaxation and Genetic Algorithms", IEEE Transactions on Power Systems, Vol. 15, No. 2, pp. 707-714, 2000.
4. C. L. Chen, S. C. Wang, "Branch-and-Bound Scheduling for Thermal Generating Units", IEEE Transactions on Energy Conversion, Vol. 8, No. 2, pp. 184-189, 1993.
5. G. Cote, M. A. Laughton, "Decomposition Techniques in Power System Planning: the Benders Partitioning Method", Electrical Power and Energy Systems, Vol. 1, No. 1, pp. 57-64, 1979.
6. F. Zhuang, F. D. Galiana, "Unit Commitment by Simulated Annealing", IEEE Transactions on Power Systems, Vol. 5, No. 1, pp. 311-318, 1990.
7. A.H. Mantawy, Y.L. Abdel-Magid, S.Z. Selim, "Unit Commitment by Tabu Search, IEE Proceedings - Generation, Transmission and Distribution, Vol. 145, No. 1, pp. 56-64, 1998.
8. S.A. Kazarlis, A.G. Bakirtzis, V. Petridis, "A Genetic Algorithm Solution to the Unit Commitment Problem", IEEE Transactions on Power Systems, Vol. 11, No. 1, pp. 83-92, 1996.
9. A. Rudolf, R. Bayrleithner, "A Genetic Algorithm for Solving the Unit Commitment Problem of a Hydro-Thermal Power System", IEEE Transactions on Power Systems, Vol. 14, No. 4, pp. 1460-1468, 1999.
10. J. Valenzuela, A. E. Smith, "A Seeded Memetic Algorithm for Large Unit Commitment Problems", Journal of Heuristics, Vol. 8, pp. 173-195, 2002.
11. G. Dudek, "Unit Commitment by Genetic Algorithm with Specialized Search Operators", Electric Power Syst. Res., Vol.72, No.3, pp.299-308, Elsevier, 2004.
12. A. Wood, B. Wollenberg, *Power Generation, Operation, and Control*, Wiley, 1996.
13. K. V. Price, R. M. Storn, J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, Springer, 2005.
14. Saramourtsis, Damousis, A. G. Bakirtzis, "Genetic Algorithm Solution to the Economic Dispatch Problem", IEE Proceedings-C, Vol. 141, No. 4, pp. 377-382, 1996.
15. J. Maturana, M. C. Riff, "Solving the Short-Term Electrical Generation Scheduling Problem by an Adaptive Evolutionary Approach", European J. of Operational Research, Vol. 179, pp. 677-691, 2007.
16. N. P. Padhy, "Unit Commitment - A Bibliographical Survey", IEEE Transactions on Power Systems, Vol. 19, No. 2, pp. 1196-2005, 2004.
17. G. Pampara , A.P. Engelbrecht, N. Franken, "Binary Differential Evolution", IEEE Congress on Evolutionary Computation, pp. 1873-1879, 2006.
18. T. Gong, A. Tuson, "Differential Evolution for Binary Encoding" 11th Online World Conference on Soft Computing in Industrial Applications, 2006.
19. A. E. Eiben, J. E. Smith, *Introduction to Evolutionary Computing*, Springer, 2003.