Unit Commitment Problem Solution Using Shuffled Frog Leaping Algorithm

Javad Ebrahimi, *Student Member, IEEE*, Seyed Hossein Hosseinian, and Gevorg B. Gharehpetian, *Senior Member, IEEE*

***Abstract*—A new evolutionary algorithm known as the shuffled frog leaping algorithm is presented in this paper, to solve the unit commitment (UC) problem. This integer-coded algorithm has been developed to minimize the total energy dispatch cost over the scheduling horizon while all of the constraints should be satisfied. In addition, minimum up/down-time constraints have been directly coded not using the penalty function method. The proposed algorithm has been applied to ten up to 100 generating units, considering one-day and seven-day scheduling periods. The most important merit of the proposed method is its high convergence speed. The simulation results of the proposed algorithm have been compared with the results of algorithms such as Lagrangian relaxation, genetic algorithm, particle swarm optimization, and bacterial foraging. The comparison results testify to the efficiency of the proposed method.**

***Index Terms*—Economic dispatch, generation scheduling, optimization techniques, shuffled frog leaping algorithm, unit commitment.**

NOMENCLATURE

N	Number of units.
T	Scheduling horizon.
D^t	System load demand at hour t.
R^t	System reserve at hour t.
C	Number of operating cycles for each unit.
T_i^c	Duration of operating cycle c for unit i.
$u_i(t)$	Operation status of unit i at hour t ($1 =$ ON and $0 =$ OFF).
P_i^t	Output power of ith unit at hour t.
$P_{i\max}$	Maximum output power of ith unit.
$P_{i\min}$	Minimum output power of ith unit.
P_{imax}^t	Maximum output power of ith unit at hour t.
P_{imin}^t	Minimum output power of ith unit at hour t.
MU_i	Maximum up-time limit of unit i.
MD_i	Minimum down-time limit of unit i.
RU_i	Ramp-up rate of unit i.
RD_i	Ramp-down rate of unit i.
$Hcost_i$	Hot start cost of unit i.
$Ccost_i$	Cold start cost of unit i.
$Chour_i$	Cold start hour of unit i.
SU_i	Start-up cost for unit i.
SD_i	Shutdown cost for unit i.
FC	Fuel cost.
TC	Total cost.
$Rand$	Random number generator with uniform distribution between 0 and 1.
$Round(x)$	Rounds x to the nearest integer.
$H(.)$	Unit step function.
$\Re(.)$	Unit ramp function (i.e., $x.H(x)$).

Manuscript received July 19, 2009; revised December 01, 2009, January 26, 2010, April 25, 2010, and May 18, 2010; accepted May 31, 2010. Paper no. TPWRS-00559-2009.

The authors are with the Electrical Engineering Department, Amirkabir University of Technology, Tehran, Iran (e-mail: J_ebrahimi@aut.ac.ir; Hosseinian@aut.ac.ir; grptian@aut.ac.ir).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TPWRS.2010.2052639

I. INTRODUCTION

UNIT COMMITMENT (UC) is used to schedule the operation of the generating units in order to satisfy load demand such that the total system operational cost over the scheduled horizon be minimized as subject to many system and generator operational constraints [1]. The solution to this problem implies a simultaneous solution of two subproblems: the mixed-integer nonlinear programming problem of determining the generating units to be running during each hour of the planning horizon, considering system capacity requirements; and the quadratic programming problem of optimally dispatching the forecasted load among the committed units during each specific hour of operation [2].

The exact solution to the UC problem can be obtained by complete enumeration, which is prohibitive owing to its excessive computational time requirements for realistic power systems [3]. Therefore, several solution methods have been proposed to solve the unit commitment problem [4], such as priority list (PL) [5], [6], dynamic programming (DP) [7], [8], Lagrangian relaxation (LR) [9]–[11], genetic algorithm (GA) [12]–[17], and particle swarm optimization (PSO) [18]–[23].

The PL method is fast but highly heuristic and gives schedules with relatively higher operation costs. The DP method has the advantage of being able to solve problems of a variety of sizes [8]. But it may lead to more mathematical complexity and increase in computation time, if the constraints are taken into consideration [24].

The LR method is capable of solving large-scale UC problems within short execution times [25]. It has the advantage of being easily modifiable to model characteristics of specific utilities and can be considered unit constraints relatively easy. The main disadvantage of this method is its inherent suboptimality.

Evolutionary algorithms (EAs), such as GA and PSO, are stochastic search methods. GA has been implemented by various researchers for the solution of UC problem. However, the disadvantage of this method is its long execution time, and there is no guarantee it will converge to the optimal solution.

The usual reported GA implementations on the UC problem are based on the binary coding. However, the formulation and implementation are difficult in arriving at an optimal solution. Other coding schemes such as integer [17] or floating point can be more efficient when accompanied with suitable GA operators [26].

In solving the UC problem by PSO, the particles do not require any repair strategies for satisfying the constraints without disturbing the optimum process of the PSO. As a result, the algorithm is capable of efficiently exploring the search space and generating quality solutions. The comparison results show that the PSO is more efficient than GA, which could obtain the global optimum solution more probably.

In this paper, a new integer-coded evolutionary algorithm known as shuffled frog leaping algorithm (SFLA) is used to solve the UC problem.

II. Shuffled Frog Leaping Algorithm

The SFLA is a meta-heuristic optimization method which is based on observing, imitating, and modeling the behavior of a group of frogs when searching for the location that has the maximum amount of available food [27]. SFLA, originally developed by Eusuff and Lansey in 2003, can be used to solve many complex optimization problems, which are nonlinear, nondifferentiable, and multi-modal [28]. SFLA has been successfully applied to several engineering optimization problems such as water resource distribution [29], bridge deck repairs [30], job-shop scheduling arrangement [31], and traveling salesman problem (TSP) [32]. The most distinguished benefit of SFLA is its fast convergence speed [33]. The SFLA combines the benefits of the both the genetic-based memetic algorithm (MA) and the social behavior-based PSO algorithm [34].

In SFLA, there is a population of possible solutions defined by a set of virtual frogs partitioned into different groups which are described as memeplexes, each performing a local search. Within each memeplex, the individual frogs hold ideas, which can be infected by the ideas of other frogs. After a defined number of memetic evolution steps, ideas are passed between memeplexes in a shuffling process. The local search and the shuffling process continue until the defined convergence criteria are satisfied [33], [35].

The flowchart of SFLA is illustrated in Fig. 1. In the first step of this algorithm, an initial population of P frogs is randomly generated within the feasible search space. The position of the

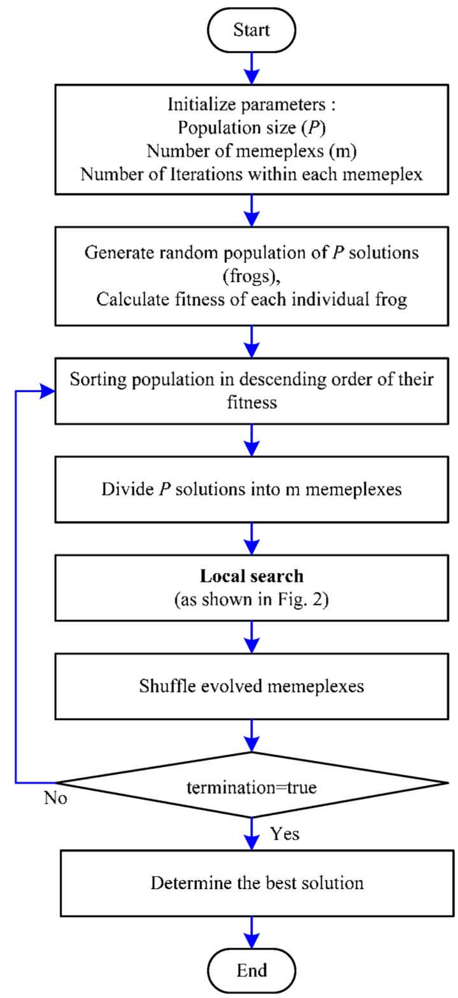


Fig. 1. Flowchart of SFLA.

ith frog is represented as $X_i = (X_{i1}, X_{i2}, \ldots, X_{iD})$, where D is the number of variables. Then, the frogs are sorted in descending order according to their fitness. Afterwards, the entire population is partitioned into m subsets referred to as memeplexes, each containing n frogs (i.e., $P = m \times n$). The strategy of the partitioning is as follows: the first frog goes to the first memeplex, the second frog goes to the second memeplex, the mth frog goes to the mth memeplex, the $(m+1)$th frog goes back to the first memeplex, and so forth. In each memeplex, the positions of frogs with the best and worst fitnesses are identified as X_b and X_w, respectively. Also the position of a frog with the global best fitness is identified as X_g. Then, within each memeplex, a process similar to the PSO algorithm is applied to improve only the frog with the worst fitness (not all frogs) in each cycle. Therefore, the position of the frog with the worst fitness leaps toward the position of the best frog, as follows:

$$D_i = Rand \times (X_b - X_w) \tag{1}$$

$$X_w^{new} = X_w^{current} + D_i \quad (D_{i\,\min} < D_i < D_{i\,\max}) \tag{2}$$

where $D_{i\,\max}$ and $D_{i\,\min}$ are the maximum and minimum step sizes allowed for a frog's position, respectively.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

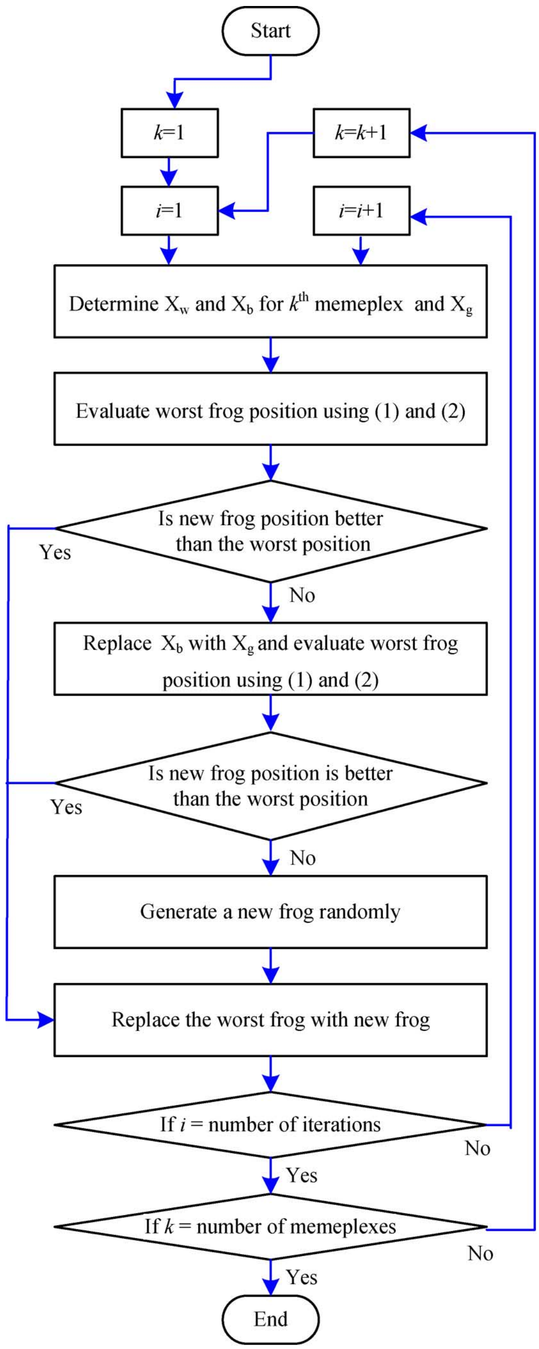


Fig. 2. Flowchart of local search.

If this process produces a better solution, it will replace the worst frog. Otherwise, the calculations in (1) and (2) are repeated but X_b is replaced by X_g. If there is no improvement in this case, a new solution will be randomly generated within the feasible space to replace it. The calculations will continue for a specific number of iterations [30], [33]. Therefore, SFLA simultaneously performs an independent local search in each memeplex using a process similar to the PSO algorithm. The flowchart of local search of SFLA is illustrated in Fig. 2.

After a predefined number of memetic evolutionary steps within each memeplex, the solutions of evolved memeplexes $(X_1, \ldots, X_P)$ are replaced into new population. (new population $= \{X_k, k = 1, \ldots P\}$); this is called the shuffling process. The shuffling process promotes a global information exchange among the frogs. Then, the population is sorted in order of decreasing performance value and updates the population best frog's position X_g, repartition the frog group into memeplexes, and progress the evolution within each memeplex until the conversion criteria are satisfied. Usually, the convergence criteria can be defined as follows [36]:

- The relative change in the fitness of the global frog within a number of consecutive shuffling iterations is less than a pre-specified tolerance.
- The maximum predefined number of shuffling iteration has been obtained.

III. UC Problem Formulation

The total operating cost of the UC problem is expressed as the sum of fuel costs, start-up costs, and shutdown costs of the generating units [3]. The fuel cost is the major component of the operating cost, which is normally modeled by a quadratic input/output curve. The start-up costs are expressed by an exponential or linear function of time [3]. In this paper, start-up costs are modeled as a two-valued (hot start/cold start) staircase function [12].

The system and unit constraints, which must be satisfied during the optimization process, are as follows:

- The unit initial operation status.
- The minimum up and down times: The minimum up/down-time constraints indicate that a unit must be ON/OFF for a minimum time before it can be shut down or restarted, respectively. These constraints are expressed by the following equations:

$$\begin{cases} T_i^c \geq MU_i, & \text{if } T_i^c \geq 0 \\ -T_i^c \geq MD_i, & \text{if } T_i^c < 0 \end{cases} \tag{3}$$

where T_i^c is a sign integer that represents the continuous ON/OFF status duration of the cth cycle of unit i. The sum of T_i^c for each unit must be equal to the scheduling horizon, i.e.,

$$\sum_{c=1}^{C} |T_i^c| = T. \tag{4}$$

- The upper and lower limits of the ith generation unit, as follows:

$$P_{i\min}^t < P_i^t < P_{i\max}^t. \tag{5}$$

- The ramp up and down rates: Considering the response rate constrains of the unit, the power generation of the unit is limited by the following time-dependent operating limits:

$$P_{i\max}^t = \min\left\{P_{i\max}, P_i^{t-1} + \tau.RU_i\right\} \tag{6}$$
$$P_{i\min}^t = \max\left\{P_{i\min}, P_i^{t-1} - \tau.RD_i\right\} \tag{7}$$

where τ is equal to 60 min and it is the UC time step.

- The power balance of the power system is presented by the following equation:

$$\sum_{i=1}^{N} u_i(t).P_i^t = D^t \quad t = 1, \ldots, T. \tag{8}$$

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I
CONFIGURATION OF X FOR TEN-UNIT SYSTEM

Unit 1					Unit 2					...	Unit 10				
X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	...	X_{46}	X_{47}	X_{48}	X_{49}	X_{50}
T_1^1	T_1^2	T_1^3	T_1^4	T_1^5	T_2^1	T_2^2	T_2^3	T_2^4	T_2^5	...	T_{10}^1	T_{10}^2	T_{10}^3	T_{10}^4	T_{10}^5
24	0	0	0	0	24	0	0	0	0	...	-11	1	-7	1	-4

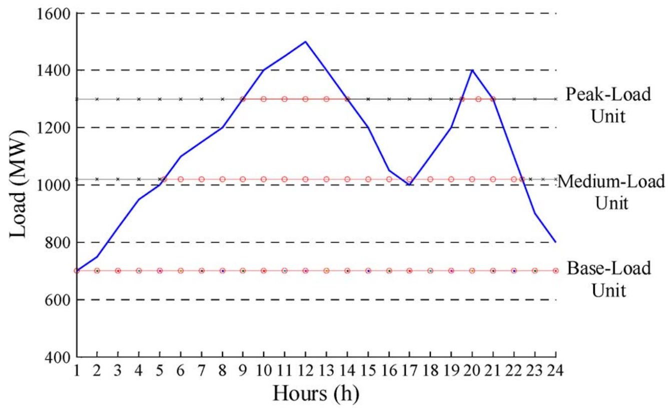


Fig. 3. Base-load, medium-load, and peak-load operating cycles.

- The spinning reserve (10-min) of the power system: The spinning reserve is the total amount of the real power generation available from all operated units minus the present load. It has to satisfy the following equation:

$$\sum_{i=1}^{N} u_i(t).P_{i\max r}^t \geq D^t + R^t \quad t = 1, \ldots, T \tag{9}$$

where $P_{i\max r}^t$ is the 10-min maximum response rate constrained power generation of the ith unit, and is defined by (6) with $\tau = 10$.

IV. APPLICATION OF SFLA ON UC PROBLEM

A. Frog (Solution) Definition

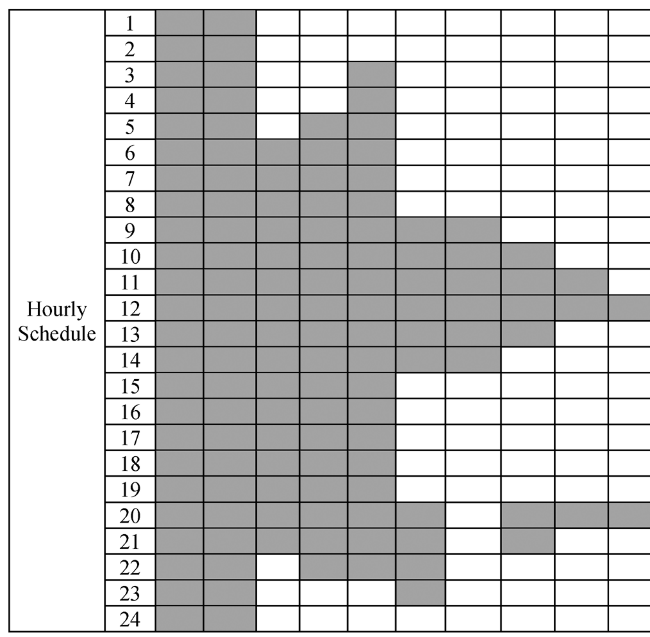
In the integer coded SFLA, the frog position (X) consists of a sequence of integer numbers, representing the sequence of the ON/OFF cycle durations of each unit during the UC horizon. A positive integer in the X represents the duration of continuous unit operation (ON status), while a negative integer represents the duration of continuous reservation (OFF status) of the unit. The number of a unit's "ON/OFF" cycles during the UC horizon depends on the number of load peaks during the UC horizon and the sum of the minimum up and down times of the unit [17]. Fig. 3 shows a daily load profile with two load peaks used to determine the number of ON/OFF cycles of units.

The numbers of ON/OFF cycles of the base, medium, and peak load units are equal to 2, 3, and 5, respectively. Therefore, the number of ON/OFF cycles of generating units is usually small (1 to 5 ON/OFF cycles per day). The reduction of cycles of base and medium units may restrict the search space of the optimization problem and this may lead to suboptimal solutions [17]. To overcome this problem in the proposed algorithm, the number of cycles of units per scheduling is the same and equal to the number of the cycle of peak load units (i.e., 5). For Y-day scheduling, C is equal to $Y \times 5$. Therefore, each

TABLE II
UNIT COMMITMENT SCHEDULE FOR 24 h

	Unit	1	2	3	4	5	6	7	8	9	10
Cycles	1	24	24	-5	-4	-2	-8	-8	-9	-10	-11
	2	0	0	16	18	20	6	6	4	2	1
	3	0	0	-3	-2	-2	-5	-10	-6	-7	-7
	4	0	0	0	0	0	4	0	2	1	1
	5	0	0	0	0	0	-1	0	-3	-4	-4

↓



solution consists of $N \times Y \times 5$ variables for Y-day scheduling and presents the operation schedule of N units for $Y \times 24$ hours. Table I gives the configuration of X for ten-unit power system and Table II lists the unit commitment schedule for 24 h.

B. Initial Population of SFLA

The generation of the initial population of SFLA is discussed in this section. The duration of the unit i operation first cycle, T_i^1, is initialized so that the unit continues the operating mode (ON/OFF) of the last cycle of the previous scheduling day for at least as many hours as required to satisfy the minimum up/down-time constraints [17]:

$$T_i^1 = \begin{cases} +Rand\left(\max\left(0, MU_i - T_i^0\right), T\right), & \text{if } T_i^0 > 0 \\ -Rand\left(\max\left(0, MD_i + T_i^0\right), T\right), & \text{if } T_i^0 < 0 \end{cases} \tag{10}$$

where T_i^0 is the duration of last cycle of the previous scheduling day.

For $c < C$, the operation duration of the cth cycle of unit i, T_i^c, is calcula[illegible]onsidering the minimum up and down-time

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

constraints of the unit, the UC horizon and the duration of the $(c-1)$ prior cycles of the unit's operation.

For $T_i^{c-1} < 0$, cycle c is in ON mode with duration determined, as follows:

$$T_i^c = \begin{cases} +Rand\left(MU_i, RT_i^{c-1}\right), & \text{if } \left(RT_i^{c-1} > MU_i\right) \\ +RT_i^{c-1}, & \text{otherwise.} \end{cases} \tag{11}$$

For $T_i^{c-1} > 0$, cycle c is in OFF mode with duration determined by the following equation:

$$T_i^c = \begin{cases} -Rand\left(MD_i, RT_i^{c-1}\right), & \text{if } \left(RT_i^{c-1} > MD_i\right) \\ -RT_i^{c-1}, & \text{otherwise} \end{cases} \tag{12}$$

where RT_i^{c-1} expresses the scheduling time remaining after allocation of the first $(c-1)$ cycles and calculated by the following equation:

$$RT_i^{c-1} = T - \sum_{p=1}^{c-1} \left|T_i^p\right|. \tag{13}$$

Considering randomly generated cycle durations, the entire scheduling period is covered with the first $c < C$ operating cycles in some cases. Therefore, the remaining $(c+1,\ldots,C)$ cycles are assigned to zero. Then, the remaining positions of X for unit i are filled with zeros.

After determination of the initial population, the unit minimum up and down-time constraints are automatically satisfied. Therefore, the penalty functions should not be used in the SFLA fitness function [17].

C. Leaping of Worst Solution

In each memeplex, the solution with the worst fitness, X_w, is adjusted by adding a vector $(D_i = Rand \times (X_b - X_w))$ to it. This approach leads to the sum of values of T_i^c for each unit, which is not equal to the scheduling horizon. Therefore, the operating cycles of each unit of new X_w should be corrected, as follows:

$$\left(T_i^1,\ldots,T_i^C\right) = \frac{T}{\sum_{k=1}^{C}\left|T_i^k\right|} \cdot \left(T_i^1,\ldots,T_i^C\right) \quad i=1,2,\ldots,N. \tag{14}$$

The function $Rand$ generates a random number between 0 and 1. As a result, the parameters of new X_w are not integer. But in the UC problem, parameters should be only integer. Therefore, the parameters of new X_w must be converted to integer numbers, as follows:

$$X'_w = Round(New\ X_w) \tag{15}$$

while X'_w is a new solution with integer parameters. It should be noted that $Round$ function will change the values of parameters of new X_w. Therefore, the sum of the values of T_i^c for each unit is not equal to the scheduling horizon. The duration of the last nonzero cycle (T_i^l) of each unit should be changed, by using the following equation:

$$T_i^l = T - \sum_{k=1}^{l-1} \left|T_i^k\right|, \quad i = 1,2,\ldots,N. \tag{16}$$

D. Satisfying Minimum Up and Down-Time Constraints

After generation of the new solution, the minimum up and down-time constraints are checked without using any penalty function. Suppose that the unit i in cycle c was in operation less than its minimum up/down-times. In order to satisfy the time constraint, first, the minimum up/down-time constraint of the cycle c should be considered. In this case, the duration of the cycle c will be equal to the minimum up/down-time. Then, the operation of the cycle $c+1$ should be changed so that the sum of T_i^c for the unit i become equal to the scheduling horizon. This procedure is presented in the following paragraphs.

The duration of the operation first cycle of unit i is checked with respect to the duration of the last cycle of the previous scheduling day and minimum up and down-time constraints of unit i.

For $T_i^1 > 0$, if $T_i^1 < \max(0, MU_i - T_i^0)$, then the duration of cycles 1 and 2 of unit i are changed, as follows:

$$\begin{cases} T_i^2 = T_i^2 - T_i^1 + \max\left(0, MU_i - T_i^0\right) \\ T_i^1 = \max\left(0, MU_i - T_i^0\right). \end{cases} \tag{17}$$

For $T_i^1 < 0$, if $-T_i^1 < \max(0, MD_i + T_i^0)$, then the duration of cycles 1 and 2 of unit i are changed, as follows:

$$\begin{cases} T_i^2 = T_i^2 - T_i^1 - \max\left(0, MD_i + T_i^0\right) \\ T_i^1 = \max\left(0, MD_i + T_i^0\right). \end{cases} \tag{18}$$

The duration of cycles $c = 2,\ldots,C-1$ of unit i are checked considering the minimum up and down-time constraint of unit i.

For $T_i^c > 0$, if $T_i^c < MU_i$, then the duration of cycles c and $c+1$ of unit i are changed, as follows:

$$\begin{cases} T_i^{c+1} = T_i^{c+1} - T_i^c + MU_i \\ T_i^c = MU_i. \end{cases} \tag{19}$$

For $T_i^c < 0$, if $-T_i^c < MD_i$, then the duration of the cycles c and $c+1$ of unit i are changed, as follows:

$$\begin{cases} T_i^{c+1} = T_i^{c+1} - T_i^c - MD_i \\ T_i^c = MD_i. \end{cases} \tag{20}$$

It should be noted that after leaping the worst solution and satisfying time constraints, an economic dispatch (ED) should be carried out in each hour of scheduling horizon for on-state units. Then, the fitness function will be calculated.

E. Fitness Function Computation

The objective function of SFLA has two terms. The first term is the total operation cost over scheduling horizon and the second term is the penalty function that penalizing the violation of system constraints. All the generators are assumed to be connected to the same bus supplying the total system demand. Therefore, the network constraints are not considered. In the first step, an ED should be performed for the scheduling horizon. It is an important part of UC [3]. Its goal is to minimize the total generation cost of a power system for each hour while satisfying constraints. The penalty functions of reserve and generation constraints are used to solve ED for the scheduling

TABLE III
OPERATOR DATA FOR TEN-UNIT SYSTEM

	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	Unit 6	Unit 7	Unit 8	Unit 9	Unit 10
$P_{i\,\max}$	455	455	130	130	162	80	85	55	55	55
$P_{i\,\min}$	150	150	20	20	25	20	25	10	10	10
A_i	1000	970	700	680	450	370	480	660	665	670
B_i	16.19	17.26	16.60	16.50	19.70	22.26	27.74	25.92	27.27	27.79
C_i	0.00048	0.00031	0.002	0.00211	0.00398	0.00712	0.00079	0.00413	0.00222	0.00173
MU_i	8	8	5	5	6	3	3	1	1	1
MD_i	8	8	5	5	6	3	3	1	1	1
$Hcost_i$	4500	5000	550	560	900	170	260	30	30	30
$Ccost_i$	9000	10000	1100	1120	1800	340	520	60	60	60
$Chour_i$	5	5	4	4	4	2	2	0	0	0
ini state	8	8	-5	-5	-6	-3	-3	-1	-1	-1

TABLE IV
LOAD DEMAND FOR 24 h

Hour [h]	1	2	3	4	5	6	7	8	9	10	11	12
Demand [MW]	700	750	850	950	1000	1100	1150	1200	1300	1400	1450	1500
Hour [h]	13	14	15	16	17	18	19	20	21	22	23	24
Demand [MW]	1400	1300	1200	1050	1000	1100	1200	1400	1300	1100	900	800

horizon. As written in (21), a quadratic polynomial with coefficients A_i, B_i, and C_i is used to model the fuel cost function of the generation of P_i in the ith unit at the tth hour:

$$FC_i\left(P_i^t\right) = A_i + B_i.P_i^t + C_i.\left(P_i^t\right)^2. \tag{21}$$

The calculated power of each unit (P_i) from ED is used to calculate the fitness of each solution in the UC problem.

The start-up/shutdown costs are calculated as follows, respectively:

$$SU_T = \sum_{i=1}^{N}\sum_{c=2}^{C} \mathrm{H}\left(T_i^c\right).SU_i\left(-T_i^{c-1}\right) \tag{22}$$

$$SD_T = \sum_{i=1}^{N}\sum_{c=2}^{C} \left[1 - H\left(T_i^c\right)\right].SD_i. \tag{23}$$

The start-up cost depends on the instant that the unit has been switched off prior to start-up

$$SU_i\left(-T_i^{c-1}\right) = \begin{cases} H\cos t_i, & \text{if } \left(MD_i - T_i^{c-1}\right) \le Chour_i \\ C\cos t_i, & \text{if } \left(MD_i - T_i^{c-1}\right) > Chour_i. \end{cases} \tag{24}$$

The total operation cost over the scheduling horizon is expressed by the following equation:

$$TC = \sum_{t=1}^{T}\sum_{i=1}^{N}\left(FC_i\left(P_i^t\right).u(t)\right) + SU_T + SD_T. \tag{25}$$

The penalty function has two terms. The first term is used to penalize sPinning reserve constraint violations modeled by (26). The second term is used to penalize excessive capacity by (27):

$$\prod_{res} = \omega.\sum_{t=1}^{T}\frac{1}{D^t}.\Re\left(\left(D^t + R^t\right) - \sum_{i=1}^{N} u_i(t).P_{i\max r}^t\right) \tag{26}$$

$$\prod_{cap} = \omega.\sum_{t=1}^{T}\frac{1}{D^t}.\Re\left(\sum_{i=1}^{N} u_i(t).P_{i\min}^t - D^t\right) \tag{27}$$

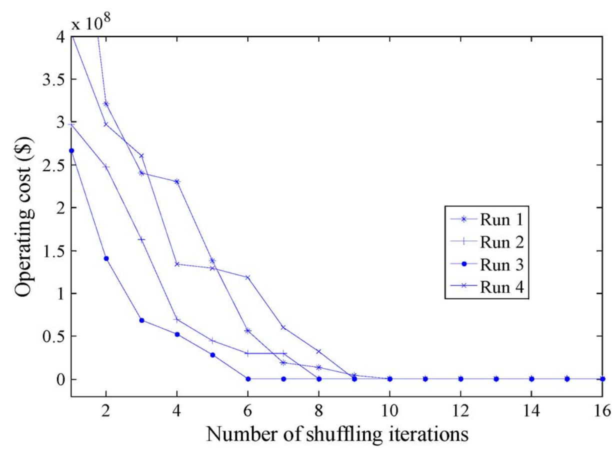


Fig. 4. Convergence of SFLA.

where ω depends on the maximum operating cost of the system over the scheduling period [17], as follows:

$$\omega = \alpha.T.\sum_{i=1}^{N} FC_i(P_{i\max}) \tag{28}$$

where α is a constant number. The overall objective of SFLA is to minimize the following fitness function subject to a number of system and unit constraints:

$$fitness = TC + \prod \tag{29}$$

where $\prod$ is equal to $\prod_{res} + \prod_{cap}$.

V. SIMULATION RESULTS

A. One-Day Scheduling

The SFLA has been tested on the ten-, 20-, 40-, 60-, 80-, and 100-unit systems over a scheduling period of 24 h. The data of a ten-unit system and load are listed in Tables III and IV, respectively [12]. For the 20-unit system, the data of ten-unit system have been duplicated and the load data doubled. The same procedure has been applied to after test system. The sPinning reserve is assumed to be 10% of the load demand.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE V
OPERATION COSTS AND SCHEDULE FOR 24 h

Hour	Power generation of units (MW) 1	2	3	4	5	6	7	8	9	10	Available reserve (%)	Generation cost ($)	Start-up cost ($)
1	455	245	0	0	0	0	0	0	0	0	30.00	13683.13	0
2	455	295	0	0	0	0	0	0	0	0	21.33	14554.50	0
3	455	370	0	0	25	0	0	0	0	0	26.12	16809.45	900
4	455	455	0	0	40	0	0	0	0	0	12.84	18597.67	0
5	455	390	0	130	25	0	0	0	0	0	20.20	20020.02	560
6	455	360	130	130	25	0	0	0	0	0	21.09	22387.04	1100
7	455	410	130	130	25	0	0	0	0	0	15.83	23261.98	0
8	455	455	130	130	30	0	0	0	0	0	11.00	24150.34	0
9	455	455	130	130	85	20	25	0	0	0	15.15	27251.05	860
10	455	455	130	130	162	33	25	10	0	0	10.86	30057.55	0
11	455	455	130	130	162	73	25	10	10	0	10.83	31916.06	60
12	455	455	130	130	162	80	25	43	10	10	10.80	33890.16	60
13	455	455	130	130	162	33	25	10	0	0	10.86	30057.55	0
14	455	455	130	130	85	20	25	0	0	0	15.15	27251.05	0
15	455	455	130	130	30	0	0	0	0	0	11.00	24150.34	0
16	455	310	130	130	25	0	0	0	0	0	26.86	21513.66	0
17	455	260	130	130	25	0	0	0	0	0	33.20	20641.82	0
18	455	360	130	130	25	0	0	0	0	0	21.09	22387.04	0
19	455	455	130	130	30	0	0	0	0	0	11.00	24150.34	0
20	455	455	130	130	162	33	25	10	0	0	10.86	30057.55	490
21	455	455	130	130	85	20	25	0	0	0	15.15	27251.05	0
22	455	385	0	0	145	20	25	0	0	0	12.45	22735.52	0
23	455	425	0	0	0	20	0	0	0	10	10.00	17645.36	60
24	455	345	0	0	0	0	0	0	0	0	13.75	15427.42	0
											Total	559847.70	4090

TABLE VI
COMPARISON OF BEST RESULTS OBTAINED BY DIFFERENT METHODS

Method	Total Start-up cost ($)	Total production cost ($)	Total operation cost ($)
GA	-	-	565825.00
DPSO	2095	562899.00	565804.00
HPSO	4090	559852.30	563942.30
SFLA	4090	559847.70	563937.70

TABLE VII
COMPARISON OF OPERATION COST OF SFLA WITH OTHER METHODS

Number of units	Total operation cost ($) LR	ICGA	BF	SFLA	Cost difference (%) LR vs SFLA	ICGA vs SFLA	BF vs SFLA
10	565825	566404	564842	564769	-0.19	-0.19	-0.01
20	1130660	1127244	1124892	1123261	-0.70	-0.35	-0.15
40	2258503	2254123	2246223	2246005	-0.56	-0.36	-0.01
60	3394066	3378108	3369237	3368257	-0.77	-0.29	-0.03
80	4526022	4498943	4491287	4503928	-0.49	0.11	0.28
100	5657277	5630838	5611514	5624526	-0.58	-0.11	0.23

The main parameters of SFLA have been selected as suggested in [34]. The SFLA has an initial population of 200 solutions, a set of 20 memeplexes, and ten generations within each memeplex (before shuffling).

The developed SFLA program has been carried out on a Pentium IV 2-GHz PC with a 512 Mbyte RAM (in MATLAB).

Considering the simulation results, it can be said that the optimal solution is obtained for 12th to 16th shuffling iteration for ten-unit system. Fig. 4 shows the high speed of convergence rate of SFLA for four runs for ten-unit system.

The result of the generation scheduling of the best solution of SFLA for ten-unit system is given in Table V.

Table VI lists the best solution of UC of ten-unit system obtained by SFLA, GA [12], discrete PSO (DPSO) [22], and hybrid PSO (HPSO) [23]. It is obvious that the total cost obtained by SFLA is less than that of other methods.

EAs have a stochastic nature and in different cases do not converge to the same solution. Therefore, the average of different cases is calculated for each problem. In the proposed method, the population size and the number of memeplexes are fixed for different test cases. In Table VII, the averages of the result of ten tests determined by SFLA have been compared with the results of LR, ICGA, and BF algorithm reported in [12], [17], and [30], respectively. It is obvious that the SFLA has satisfactory results in comparison with other methods.

The execution time is an important factor, too. This point has not been reported for LR [12] and ICGA [17]. But for the BF algorithm, the same PC has been used [30].

The execution times of SFLA and BF algorithms [30] for different systems have been compared in Table VIII and Fig. 5. It is obvious that the execution time of SFLA increases semi-linearly with increase of the size of UC problem. For all cases, SFLA is better than BF.

B. Seven-Day Scheduling

In the seven-day UC problem, the duration of the first cycle in each day depends on the duration of the last cycle of the previous day. For seven-day scheduling case, the data given in Tables III and IV have been used. Also, the daily load factor for seven days is listed in Table IX. For each hour, ED is carried out considering the corresponding load factor.

The operating cost of ten up to 100 unit systems for a seven-day scheduling period is compared with the result of the BF algorithm [30] in Table X. It is shown that the SFLA

TABLE VIII
COMPARISON OF EXECUTION TIMES OF SFLA WITH BF

Number of units	BF		SFLA	
	Execution time (sec)	Number of chemotactics	Execution time (sec)	Number of shuffling iterations
10	110	500	35	12
20	350	800	80	16
40	620	1000	150	22
60	1400	1200	280	28
80	3200	1500	690	38
100	5800	2000	1430	50

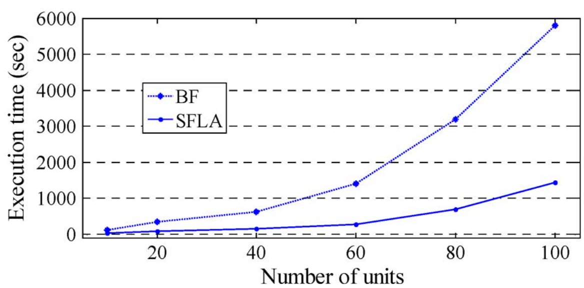


Fig. 5. Execution times for BF algorithm and SFLA.

TABLE IX
LOAD FACTOR OF EACH DAY

Day	1	2	3	4	5	6	7
Load Factor	1	0.95	0.9	0.9	0.92	0.85	0.8

TABLE X
COMPARISON OF OPERATION COST FOR SEVEN-DAY SCHEDULING

Number of units	Total operation cost ($)		Cost difference (%)	Execution time (sec)
	BF	SFLA	BF vs SFLA	SFLA
10	3529021	3518628	-0.30	210
20	7034913	6963294	-1.03	610
40	14038256	13918930	-0.86	1150
60	21098765	20772846	-1.57	1900
80	28167453	27830576	-1.21	4600
100	35257271	35058528	-0.57	9800

results in lower costs as compared to the BF algorithm in the seven-day scheduling period.

VI. CONCLUSION

This paper has proposed a new evolutionary algorithm known as SFLA to solve the UC problem. The combination of the local search with information exchange of groups results in performance improvement of SFLA. In the UC problem, the minimum up and down-time constraints have been considered during generating the feasible solutions. Therefore, there is no need to use the penalty functions method.

The efficiency of the proposed algorithm has been studied considering periods of one-day and seven-day scheduling for ten up to 100-unit systems. The proposed method has been compared with other methods. The simulation results show that the computation times and production costs of SFLA are less than other algorithms such as LR, GA, PSO, and BF.

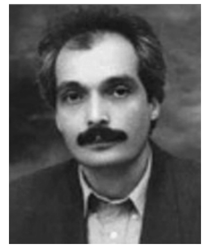
REFERENCES

[1] M. Shahidehpour, H. Yamin, and Z. Li, *Market Operations in Electric Power Systems*. New York: Wiley, 2002.
[2] D. N. Simopoulos, S. D. Kavatza, and C. D. Vournas, “Reliability constrained unit commitment using simulated annealing,” *IEEE Trans. Power Syst.*, vol. 21, no. 4, pp. 1699–1706, Nov. 2006.
[3] A. J. Wood and B. F. Wollenberg, *Power Generation Operation and Control*. New York: Wiley, 1984.
[4] N. P. Padhy, “Unit commitment—A bibliographical survey,” *IEEE Trans. Power Syst.*, vol. 19, no. 2, pp. 1196–1205, May 2004.
[5] H. H. Happ, R. C. Johnson, and W. J. Wright, “Large scale hydro-thermal unit commitment-method and results,” *IEEE Trans. Power App. Syst.*, vol. PAS-90, no. 3, pp. 1373–1384, May/Jun. 1971.
[6] C. J. Baldwin, K. M. Dale, and R. F. Dittrich, “A study of economic shutdown of generating units in daily dispatch,” *IEEE Trans. Power App. Syst.*, vol. PAS-78, pp. 1272–1284, 1960.
[7] C. K. Pang, G. B. Sheble, and F. Albuyeh, “Evaluation of dynamic programming based methods and multiple area representation for thermal unit commitment,” *IEEE Trans. Power App. Syst.*, vol. PAS-100, pp. 1212–1218, Mar. 1981.
[8] W. L. Snyder, Jr., H. D. Powell, Jr., and J. C. Rayburn, “Dynamic programming approach to unit commitment,” *IEEE Trans. Power Syst.*, vol. 2, no. 2, pp. 339–347, May 1987.
[9] S. Virmani, E. C. Adrian, K. Imhof, and S. Muhherjee, “Implementation of a Lagrangian based unit commitment problem,” *IEEE Trans. Power Syst.*, vol. 4, no. 4, pp. 1373–1380, Nov. 1989.
[10] K. Aoki, M. Itoh, T. Satoh, K. Nara, and M. Kanezashi, “Optimal long-term unit commitment in large scale systems including fuel constrained thermal and pumped-storage hydro,” *IEEE Trans. Power Syst.*, vol. 4, no. 3, pp. 1065–1073, Aug. 1989.
[11] F. N. Lee, “A fuel-constrained unit commitment method,” *IEEE Trans. Power Syst.*, vol. 4, no. 3, pp. 691–698, Aug. 1989.
[12] S. A. Kazarlis, A. G. Bakirtzis, and V. Petridis, “A genetic algorithm solution to the unit commitment problem,” *IEEE Trans. Power Syst*, vol. 11, no. 1, pp. 83–92, Feb. 1996.
[13] A. Rudolf and R. Bayrleithner, “A genetic algorithm for solving the unit commitment problem of a hydro-thermal power systems,” *IEEE Trans. Power Syst.*, vol. 14, no. 4, pp. 1460–1468, Nov. 1999.
[14] H. Yang, P. Yang, and C. Huang, “A parallel genetic algorithm approach to solving the unit commitment problem: Implementation on the transputer networks,” *IEEE Trans. Power Syst.*, vol. 12, no. 2, pp. 661–668, May 1997.
[15] K. S. Swarup and S. Yamashiro, “Unit commitment solution methodology using genetic algorithm,” *IEEE Trans. Power Syst.*, vol. 17, no. 1, pp. 87–91, Feb. 2002.
[16] J. M. Arroyo and A. J. Conejo, “A parallel repair genetic algorithm to solve the unit commitment problem,” *IEEE Trans. Power Syst.*, vol. 17, no. 4, pp. 1216–1224, Nov. 2002.
[17] I. G. Damousis, A. G. Bakirtzis, and P. S. Dokopoulos, “A solution to the unit commitment problem using integer-coded genetic algorithm,” *IEEE Trans. Power Syst.*, vol. 19, no. 2, pp. 1165–1172, May 2004.
[18] W. Xiong, M. J. Li, and Y. Cheng, “An improved particle swarm optimization algorithm for unit commitment,” in *Proc. ICICTA 2008*.
[19] V. S. Pappala and I. Erlich, “A new approach for solving the unit commitment problem by adaptive particle swarm optimization,” in *Proc. IEEE Power and Energy Soc. General Meeting*, Jul. 2008, pp. 1–6.
[20] P. Bajpai and S. N. Singh, “Fuzzy adaptive particle swarm optimization for bidding strategy in uniform price spot market,” *IEEE Trans. Power Syst.*, vol. 22, no. 4, pp. 2152–2160, Nov. 2007.
[21] A. Y. Saber, T. Senjyu, N. Urasaki, and T. Funabashi, “Unit commitment computation—A novel fuzzy adaptive particle swarm optimization approach,” in *Proc. IEEE PSCE*, Nov. 2006, pp. 1820–1828.
[22] Z. L. Gaing, “Discrete particle swarm optimization algorithm for unit commitment,” in *Proc. IEEE Power Eng. Soc. General Meeting*, Jul. 2003, vol. 1, pp. 13–17.
[23] T. O. Ting, M. V. C. Rao, and C. K. Loo, “A novel approach for unit commitment problem via an effective hybrid particle swarm optimization,” *IEEE Trans. Power Syst.*, vol. 21, no. 1, pp. 411–418, Feb. 2006.
[24] Z. Ouyang and S. M. Shahidehpour, “A multi-stage intelligence system for unit commitment,” *IEEE Trans. Power Syst.*, vol. 7, no. 2, pp. 639–646, May 1992.
[25] N. Zendehdel, A. Karimpour, and M. Oloomi, “Optimal unit commitment using equivalent linear minimum up and down time constraints,” in *Proc. 2nd IEEE PECon 08*, 2008.
[26] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. New York: Springer-Verlag, 1996.

[27] M. M. Eusuff, K. E. Lansey, and F. Pasha, "Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization," *Eng. Optimiz.*, vol. 38, no. 2, pp. 129–154, 2006.
[28] X. Zhang, X. Hu, G. Cui, Y. Wang, and Y. Niu, "An improved shuffled frog leaping algorithm with cognitive behavior," in *Proc. 7th World Congr. Intelligent Control and Automation*, 2008.
[29] M. Eslamian, S. H. Hosseinian, and B. Vahidi, "Bacterial foraging-based solution to the unit-commitment problem," *IEEE Trans. Power Syst.*, vol. 24, no. 3, pp. 1478–1488, Aug. 2009.
[30] H. Elbehairy, E. Elbeltagi, and T. Hegazy, "Comparison of two evolutionary algorithms for optimization of bridge deck repairs," *Computer-Aided Civil Infrastr. Eng.*, vol. 21, pp. 561–572, 2006.
[31] A. Rahimi-Vahed and A. H. Mirzaei, "Solving a bi-criteria permutation flow-shop problem using shuffled frog-leaping algorithm," in *Soft Computing*. New York: Springer-Verlag, 2007.
[32] X.-H. Luo, Y. Yang, and X. Li, "Solving TSP with shuffled frog-leaping algorithm," in *Proc. ISDA 2008*, vol. 3, pp. 228–232.
[33] E. Elbeltagi, T. Hegazy, and D. Grierson, "Comparison among five evolutionary-based optimization algorithms," *Adv. Eng. Informat.*, vol. 19, no. 1, pp. 43–53, 2005.
[34] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. IEEE Conf. Neural Networks*, 1995, vol. 4, pp. 1942–1948.
[35] M. M. Eusuff and K. E. Lansey, "Optimization of water distribution network design using the shuffled frog leaping algorithm," *J. Water Resour. Plan. Manage.*, vol. 129, no. 3, pp. 210–225, 2003.
[36] T. H. Huynh, "A modified shuffled frog leaping algorithm for optimal tuning of multivariable PID controllers," in *Proc. ICIT 2008*, pp. 1–6.

Javad Ebrahimi (S'10) was born in Isfahan, Iran, in 1986. He received the B.Sc. degree in electrical engineering from Tabriz University, Tabriz, Iran, in 2008. He is currently pursuing the M.Sc. degree in the Electrical Engineering Department at Amirkabir University of Technology (AUT), Tehran, Iran.

His research interests include power system optimization and operation, FACTS devices, and power electronics, in particular, multilevel converters.



Seyed Hossein Hosseinian was born in 1961 in Iran. He received the B.Sc. and M.Sc. degrees from the Electrical Engineering Department of Amirkabir University of Technology (AUT), Tehran, Iran, in 1985 and 1988, respectively, and the Ph.D. degree from the Electrical Engineering Department, University of Newcastle, Newcastle upon Tyne, U.K., in 1995.

Presently, he is an Associate Professor in the Electrical Engineering Department at AUT. His special fields of interest include transient in power systems, power quality, restructuring, and deregulation in power systems.

Gevorg B. Gharehpetian (M'00–SM'08) received the B.Sc. degree in electrical engineering from Tabriz University, Tabriz, Iran, in 1987, the M.Sc. degree in electrical engineering from Amirkabir University of Technology (AUT), Tehran, Iran, in 1989, and the Ph.D. degree in electrical engineering from the University of Tehran in 1996.

He held the Assistant Professor position at AUT from 1997 to 2003, the position of Associate Professor from 2004 to 2007, and has been a Professor since 2007. His teaching and research interests include power system and transformers transients, FACTS devices, and HVDC transmission.