A Two–Level Method for Solving Power Generating Unit Commitment Problems

Hariklia A. Georgopoulou and Kyriakos C. Giannakoglou
Lab. of Thermal Turbomachines, School of Mechanical Engineering,
National Technical University of Athens, NTUA
P.O. Box 64069, Athens 157 10, Greece
Email:chgeorg@mail.ntua.gr
Tel.:(+30)-210.772.16.36 Fax: (+30)-210.772.37.89

***Abstract*—This paper is concerned with the solution of unit commitment problems by means of a two–level Evolutionary Algorithm (*EA*), handling the encoded operational states of units. The *EA* is assisted by the augmented Lagrange relaxation method to compute the optimal units' loading, for each candidate configuration. Emphasis is laid on the reduction of the CPU cost of the proposed method and, for this purpose, a two–level *EA* has been devised. At the first–preparatory level, the time units (hours) are grouped, based on heuristics. This gives rise to a coarse–grained optimization problem with a reduced number of unknowns which can readily be solved using *EAs*, without taking into consideration constraints related to units' start–up and shutdown. The so–obtained, sub–optimal and, likely, infeasible solution is then processed by the second level for further refinement. At this level, instead of optimizing the unit commitment over the whole period of time, a small number of consecutive sub–periods are formed and the corresponding optimization sub–problems are solved iteratively. For each sub–problem, an *EA* coupled with chromosome repairing and cost function penalization to account for inconsistencies between the current sub–problem solution and those computed at the adjacent sub–intervals.**

I. Introduction

Unit Commitment (*UC*) is a constrained optimization problem seeking for the optimal schedule of a number of power generating units, to cover a given power demand distribution over a period of time, at minimum cost. The total operating cost, which is usually the objective function, sums up fuel cost, unit start–up and shut–down costs, as well as any other relevant cost (such as maintenance costs, if any). Alternatively, a second objective, that of solution robustness, can also be introduced to account for a stochastically varying power demand distribution. In *UC* problems, full enumeration is the safest way to reach the global optimum; however, this cannot be used to solve industrial problems, due to the computing cost it entails.

In view of the above, various deterministic or stochastic methods have been developed to solve the combinatorial *UC* problem, [1], such as priority list, mixed integer–real linear programming (*MILP*) [8], Lagrangian relaxation [3], dynamic programming, simulated annealing [5], [6], tabu search and evolutionary algorithms (*EAs*), [4], [2]. However, the need to safely capture global optimal solutions at the minimum CPU cost is vital, since all the aforementioned methods have at least some known weaknesses. For instance, *MILP* solvers require linear approximations of the objective function, dynamic programming risks providing with a local optimum, etc.

EAs have gained particular attention since they are friendly single– and multi–objective optimization tools, suitable to handle discontinuous and multimodal objective functions. Nevertheless, they suffer from the high number of calls to the evaluation software required in order to locate the optimal solution. Also, the higher the number of design variables, the slower conventional *EAs* become. In order to overcome these weaknesses, this paper proposes a two–level *EA* customized for the solution of *UC* problems with the following main features:

- the first level acts as a low–cost preparatory phase, capable of feeding the second optimization level with a good initial solution;
- at the first level, a coarse–grained problem with relaxed constraints is solved;
- the second level handles the full problem after, however, splitting it into a small number of consecutive time intervals, on which the optimization relies on *EAs* with chromosome repairing;
- at the second level, the optimization is iterative and the objective function associated with each time sub–interval includes penalties due to possible inconsistencies with the adjacent sub–intervals' solutions.

Finally, let us stress that the proposed algorithm was inspired by the concept of hierarchical *EA*–based optimization methods, [7].

II. *UC* Problem Formulation

Assume that M power generating units must be scheduled over T time units (hours). The *i-th* unit, abbreviated to u_i, has minimum and maximum capacities $P_{min,i}$ and $P_{max,i}$, respectively. In general, $\phi_i^{(j)}$ denotes the (given or computed) value of ϕ for u_i at $t^{(j)}$, i.e. the j-*th* hour. A deterministic power demand distribution ($d^{(j)}, j \in [1,T]$, in MW) along with the states (*ON*, *OFF*, during start–up, *STUP*, or shut–down, *SHDN*) of all units at the beginning of the time interval, the duration of *STUP* and *SHDN* procedures ($\overline{T}_i$ and $\overline{\overline{T}}_i$

for u_i, respectively), the minimum time $\widehat{T}_i$ for u_i to remain in operating or non–operating condition, are inputs to the problem. Needless to say that, basic objective is to cover the power demand over the whole time interval at minimum cost.

A. Candidate solutions' encoding

Binary coding of power generating units' states, on hourly basis, is used. The binary string for each unit is formed by concatenated bits, denoted by $s_i^{(j)}$, where $s_i^{(j)}{=}1$ if u_i is *ON* and $s_i^{(j)}=0$ otherwise (*OFF* or during *SHDN* and *STUP*). This coding is suitable for evolutionary optimization. Since $s_i^{(j)}=0$ encodes three possible operating states, one may distinguish among them by taking previous or next u_i states into consideration.

The hourly load of each unit (denoted by $x_i^{(j)}, \frac{P_{min,i}}{P_{max,i}} \leq x_i^{(j)} \leq 1$) is not involved in the binary chromosome. Once $s_i^{(j)}, i \in [1,M]$ have been determined for a specific j value, the $x_i^{(j)}, i{\in}[1,M]$ values for the operating units are computed by solving an optimization problem (separately for each hour). For its solution, the augmented Lagrange multipliers method (*ALM*, [10]) is used. It is evident that for non–operating units ($s_i^{(j)}{=}0$), $x_i^{(j)}{=}0$ by default.

B. Objective function

The *UC* problem objective is the minimization of the total operating cost (*TOC*, in MW), which sums up fuel, *STUP* and *SHDN* costs. For each u_i, the cost of a single *STUP* or *SHDN* ($\overline{C}_i$ or $\overline{\overline{C}}_i$, respectively) as well as the hourly fuel consumption $F_i^{(j)}(x_i^{(j)})$, as a function of the unit load $x_i^{(j)}$, are all known. In the present applications, $F_i^{(j)}(x_i^{(j)})$ is a quadratic polynomial in $x_i^{(j)}$. Based on the aforementioned definitions, *UC* can be envisaged as an optimization problem associated with the minimization of an objective function, defined as follows:

$$TOC{=}\sum_{j=1}^{T}\sum_{i=1}^{M} OC_i^{(j)}{+}\sum_{j=1}^{T}\Phi(\Delta d^{(j)}) \tag{1}$$

where $OC_i^{(j)}$ (in MW) is the operating cost of u_i at $t^{(j)}$ and $\Phi(\Delta d^{(j)})$ are penalty terms activated if the given demand is not exactly satisfied. The operating cost of u_i at $t^{(j)}$ depends on the u_i's state and load $x_i^{(j)}$and is given by

$$OC_i^{(j)}{=}\begin{cases} a_i\left(x_i^{(j)}\right)^2{+}b_i x_i^{(j)}{+}c_i{+}\overline{C}_i{}', & s_i^{(j)}{=}1 \\ \overline{\overline{C}}_i{}', & s_i^{(j)}{=}0 \end{cases} \tag{2}$$

where $(a_i,b_i,c_i), \forall\ i{\in}[1,M]$ are known coefficients and

$$\overline{C}_i{}' = \begin{cases} 0, & s_i^{(j-1)}{=}1 \\ \overline{C}_i, & s_i^{(j-1)}{=}0 \end{cases} \quad \overline{\overline{C}}_i{}' = \begin{cases} 0, & s_i^{(j-1)}{=}0 \\ \overline{\overline{C}}_i, & s_i^{(j-1)}{=}1 \end{cases} \tag{3}$$

Thus far, a single–objective problem formulation has been set–up. However, the demand distribution is, in fact, stochastic and follows a probability distribution. In order to take into account uncertainties related to the power demand, the same problem can be handled as a two–objective one, where the second objective quantifies the risk of not covering "reasonable" demand variations. The new objective is expressed as the possibility the power demand (at any time) falls outside the range covered by the current configuration. This range is defined by summing up the minimum/maximum capacities of the currently operating units. A normal distribution is associated with each power demand value (i.e. $d^{(j)}$is considered to be the average demand value). Since the normal cumulative function $\mathcal{Q}$ gives the probability of a standard normal variate to be found inside a desired interval, the following expression gives the failure risk $\mathcal{R}^{(j)}$:

$$\begin{gathered} \mathcal{R}^{(j)}{=}1{-}\mathcal{Q}^{(j)}, \forall j{\in}[1,T] \\ \mathcal{Q}^{(j)}{=}\frac{1}{\sqrt{2}\pi}\int_{\sum_{i=1}^{M} s_i^{(j)} p_{min_i}}^{\sum_{i=1}^{M} s_i^{(j)} p_{max_i}} \exp^{t^2/2}\,dt, \forall j{\in}[1,T] \end{gathered} \tag{4}$$

In view of the above, the second objective is given by:

$$\mathcal{R}{=}\sum_{j=1}^{T}\mathcal{R}^{(j)} \tag{5}$$

and could optionally be taken into account during the solution of the *UC* problem.

C. Constraints

The *UC* problem is an optimization problem subject to a set of constraints, as follows:

- All committed units must comply with their known states at the beginning of the time interval under consideration. Any encoded (binary) schedule that fails to satisfy this constraint should be repaired accordingly.
- The generated power should match the power demand, on an hourly basis. For candidate solutions that do not satisfy this requirement, the objective function should be penalized. The penalty function is the sum of hourly penalty terms $\Phi(\Delta d^{(j)})$. The latter are expressed in terms of a second degree polynomial of the deviation $\Delta d^{(j)} = |d^{(j)} {-} p^{(j)}|$, where $p^{(j)}$ is the hourly power production in MW and

$$\Phi(\Delta d^{(j)}){=}\begin{cases} e_1 {\Delta d^{(j)}}^2{+}e_2\Delta d^{(j)}{+}e_3, & \Delta d^{(j)} \neq 0 \\ 0, & \Delta d^{(j)}{=}0 \end{cases} \tag{6}$$

  where $(e_1,e_2,\ e_3)$ are user defined coefficients.
- Minimum *STUP*, *SHDN* and "state–change delay" time intervals ($\overline{T}$, $\overline{\overline{T}}$, $\widehat{T}$) should be taken into account during all transient phases. Binary strings which fail to satisfy these rules, should be repaired. Practically, bit positions are changed from 0 to 1, whenever a sequence of consecutive $0's$ is insufficient to include a *SHDN* process followed by a *STUP* one. Opposite changes (from 1 to 0) are not allowed. Chromosome repair is made according to the following scheme:

$$s_i^{j+1}{-}s_i^{j}{=}\begin{cases} 0 \\ 1 \Rightarrow s_i^{j+k}{=}1, \forall k{\in}[1, \overline{T}_i{+}\widehat{T}_i] \end{cases} \tag{7}$$

$$s_i^{j+1}-s_i^{j}=-1 \text{ iff } s_i^{j+k}=0, \forall k\in[1,\overline{\overline{T}}_i+\widehat{T}_i] \tag{8}$$

$$i\in[1,M],\ \ j\in[1,T]$$

III. The Two–level Optimization Algorithm

In a *UC* problem with a long time interval and/or many units, the encoded chromosome (MT bits) becomes too lengthy to handle it by means of *EAs*. Thus, the CPU cost for solving such a problem becomes almost prohibitive; also, despite the high computing cost, sub–optimal solutions may arise. In order to circumvent this problem, a two–level optimization algorithm is proposed.

The first level is a low–cost preparatory phase which, through a coarsening procedure, manages to find a ”solution” to the coarse–grained problem which, although sub–optimal and usually infeasible, is amenable to refinement at reasonable extra CPU cost. This refinement is carried out at the second optimization level. In order to reduce, as much as possible, the CPU cost of the second level task, the entire time interval is partitioned to a (usually small) number of consecutive sub–intervals. Thus, a series of optimizations over these sub–intervals, should be carried out; *EAs* are also in use for this purpose with a modified objective function to account for the matching conditions at their interfaces. The two–level algorithm is described below, in more detail:

1) First Level Optimization: The first level optimization consists of two sub–phases, namely the coarsening and the optimization of the coarsened problem. Given the power demand distribution, the coarsening procedure requires, for each time unit (hour), a call to the Mixed Integer Linear Programming software (*MILP*, branch and bound technique), which is capable of determining the units which should be up in order to cover the demand. Needless to say that, each one of the T calls to this software is extremely fast and that it does not account any restriction related to the preceding states of units. Practically, the objective function is nothing more than the fuel cost for $t^{(j)}, j\in[1,T]$, penalized by $\Phi(\Delta d^{(j)})$, as previously described. Since an optimization method for linear problems is used and the objective function is a quadratic one, the latter is linearized. In general, such a linearization is harmless, given the role of the first level optimization, which is nothing more than to provide a good initial solution to the second level.

Having obtained ”indicative” solutions for the T hours, the coarsening procedure follows. Consecutive time units, which are given the same solution during the previous sub–phase, are grouped together to form a coarser time unit. Additional grouping is possible and should be carried out (by grouping together consecutive time units or already formed groups of them, which are given ”similar” solutions) aiming at the desirable coarse problem size. Practically, the user defines the upper bound of the number of coarse time units and, then, heuristics are employed. When this preparatory or coarsening step comes to an end, T_c ($T_c \ll T$) coarse time units, generally with different size each, have been defined.

The second and final sub–phase, within the first level, is concerned with the optimization of the coarse–grained problem. This optimization is based on *EAs* with binary encoding (handling an MT_c bit string) and without considering constraints related to the transient phases (*STUP* or *SHDN*). The small bit string size makes this computation very fast. In general, the final solution might be infeasible since, as mentioned before, constraints have not been taken into account.

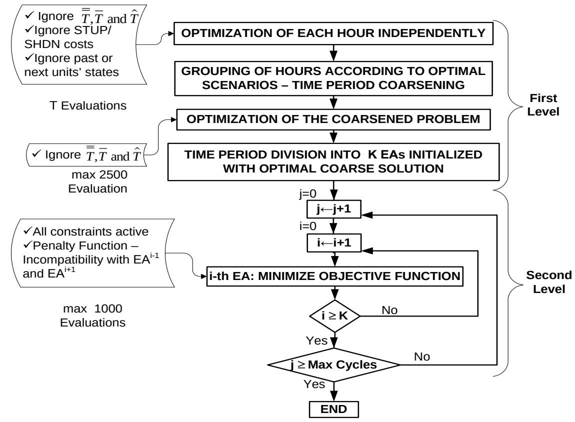


Fig. 1. Flowchart of the proposed algorithm.

2) Second Level Optimization: The optimal schedule computed at the first level for the coarsened problem is first expanded to cover the entire interval of time (T time units). At this stage and according to what has already been presented in a previous section of this paper, the optimal solution should be repaired to account for the *STUP*, *SHDN* and ”state–change delay” constraints. The outcome of the repairing phase is probably a worse, though feasible, solution to the (real) problem. This is the good starting solution that should be injected into the *EA* algorithm, which will undertake the optimization at the first level.

When the *UC* problem is handled with a stochastic demand distribution (i.e. as a two–objective problem) the outcome of the first level is the Pareto front solutions of the coarsened problem which are to be transferred to the second level.

Next step is to partition the entire time interval to K sub–intervals of the same or almost the same size. For each sub–interval, an *EA* with a modified objective function, in the sense that this additionally takes into account the matching with the previous and next sub–interval solutions should be used. The optimization is iterative and terminates when all matching conditions, at the interfaces between sub–intervals, are satisfied. By the way of example, the *UC* problem over a period of 72 hours could be solved by splitting it into 3 sub–problems with 24 hours each, or 6 sub–problems with 12 hours each and so forth. The bit string used by the *EAs* employed over each sub–interval consists of TM/K bits. A schematic representation of the proposed algorithm is given in Figure 1.

IV. Case Studies

The proposed two–level *UC* optimization algorithm was tested on two problems. The core optimization tool within the

TABLE I
CASE I: PROBLEM DEFINITION.

i	$P_{min,i}$	$P_{max,i}$	$F_i^{(j)}(x_i^{(j)})$	$\overline{C}_i$	$\overline{\overline{C}}_i$	$\overline{T}_i$	$\overline{\overline{T}}_i$	$\widehat{T}_i$	s_0
1	20	40	$80{x_i^{(j)}}^2{+}40$	120	100	2	2	1	*ON*
2	30	70	$108{x_i^{(j)}}^2{+}69x_i^{(j)}{+}63$	180	150	2	2	1	*ON*
3	50	100	$169.6{x_i^{(j)}}^2{-}5.6x_i^{(j)}{+}266$	240	200	2	2	1	*ON*
4	60	120	$180{x_i^{(j)}}^2{+}312$	240	200	2	2	1	*ON*

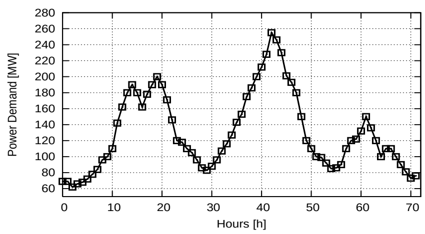
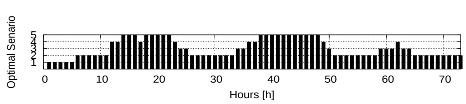


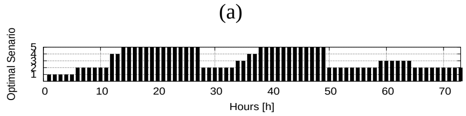
Fig. 2. Case I: Power demand distribution.

TABLE II
CASE I: CLASSIFICATION OF THE DEMAND INTO GROUPS (FIRST LEVEL).

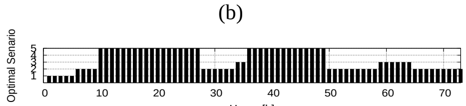
k	$Pmin_k$	$Pmax_k$	s_1	s_2	s_3	s_4
1	62	69	0	1	0	0
2	72	110	1	1	0	0
3	116	140	1	0	1	0
4	142	169	0	1	1	0
5	171	210	1	1	1	0
6	211	230	1	1	0	1
7	233	255	1	0	1	1



(a)

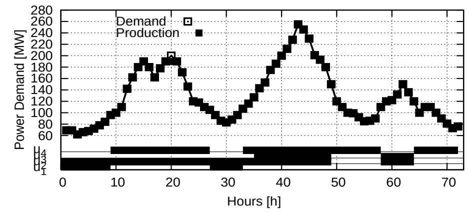


(b)

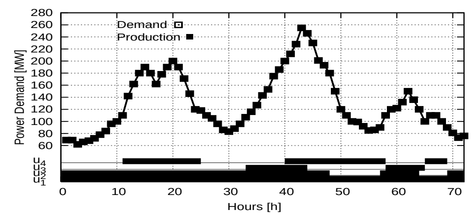


(c)

Fig. 3. Case I: The sequential coarsening procedure up to the desired maximum number of groups (in this case, this was set to 9).

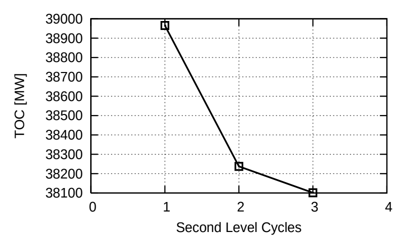


(a)



(b)

Fig. 4. Case I: Optimal solutions computed at (a) the first and (b) second optimization level the solution to the coarse problem has been expanded and repaired, accordingly.



(a)

Fig. 5. Case I: Convergence of the second level optimization. The fourth cycle solution is identical to the 3^{rd} cycle one.

customized *UC* optimization platform is software *EASY* [11], a software developed and brought to market by the National Technical University of Athens.

A. Case I

The first problem is concerned with a four unit system and a scheduling period of T=72 hours. Unit capacities, cost models and constraints for the first problem have been all included in Table I. Fig. 2 shows the power demand distribution for the T=72 hour period. According to Table I, all units are *ON* at the beginning of this time period.

At the first level, after solving 72 optimization problems through the *MILP* method, we came up with 72 hourly solutions, which can easily be classified to seven combinations of s_i, as shown in Table II. For instance, $(s_1, s_2, s_3, s_4) = (0, 1, 0, 0)$ means that only u_2 is *ON* and so forth. In the same table, one may also find the minimum and maximum power demand (in MW) associated with each one of these seven operating scenarios. The reader should follow, step–by–step, the coarsening procedure, in Fig. 3. Fig. 3(a) shows the ”optimal” scenarios computed for the 72 hours; although there are only seven different scenarios, the demand distribution varies strongly and, therefore, 19 groups of small duration (of

TABLE III
CASE II: PROBLEM DEFINITION.

i	$P_{max,i}$	$P_{min,i}$	$F_i^{(j)}(x_i^{(j)})$	$\overline{C}_i$	$\overline{\overline{C}}_i$	$\overline{T}_i$	$\overline{\overline{T}}_i$	$\widehat{T}_i$	Initial State s_0
1	60	20	$5{x_i^{(j)}}^2 - 10x_i^{(j)} + 7$	120	100	2	2	1	*ON*
2	80	30	$4.5{x_i^{(j)}}^2 - 9x_i^{(j)} + 7.5$	180	150	2	2	1	*ON*
3	100	40	$5.3{x_i^{(j)}}^2 - 9x_i^{(j)} + 8$	240	200	2	2	1	*ON*

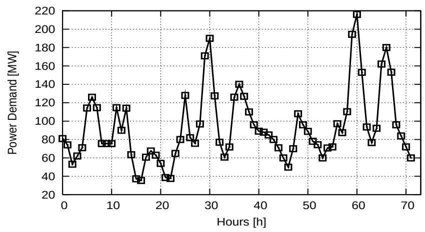


Fig. 6. Case II: Power demand distribution.

TABLE IV
CASE II: CLASSIFICATION OF THE DEMAND INTO GROUPS (FIRST LEVEL).

k	$Pmin_k$	$Pmax_k$	s_1	s_2	s_3
1	35.5	60.9	1	0	0
2	62.1	80.0	0	1	0
3	81.0	140.0	1	1	0
4	153.0	153.2	0	1	1
5	162.0	216.0	1	1	1

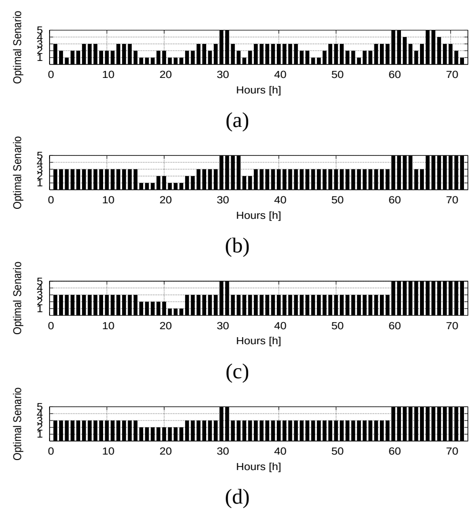


Fig. 7. Case II: The sequential coarsening procedure up to the desired maximum number of groups (in this case, this was set to 6). Note that between (a) and (b) some scenarios have been ”disappeared”.

1 up to 11 hours each) have been created. Such a coarsening is inadequate and heuristics are needed for further grouping. This is, in fact, a sequential coarsening/grouping procedure which identifies isolated groups of small size and incorporates them to the adjacent group, provided that the corresponding difference in power demand is not high enough. In our case, three sequential coarsening steps have been carried out and these are shown in figs. 3(b) and 3(c). At the end of this procedure, the entire time period was formed by to nine groups, Fig.3(c). Note that their sizes are completely different due to the power demand distribution we are dealing with. The minimum and maximum power values for each group presented in Table II have been computed according to outcome of the sequential coarsening scheme shown in Fig.3. Given this nine group coarsening, in a four unit system, a bit string of 36 digits encoded all possible solutions and was readily solved using *EAs*. Fig.4 (a) shows the optimal solution computed for the coarse–grained problem; in the same figure, the operating units which are up are marked with a thick horizontal line (bottom). The coarse problem solution fails to satisfy the power demand only at the 20^{th} hour with deficiency of power.

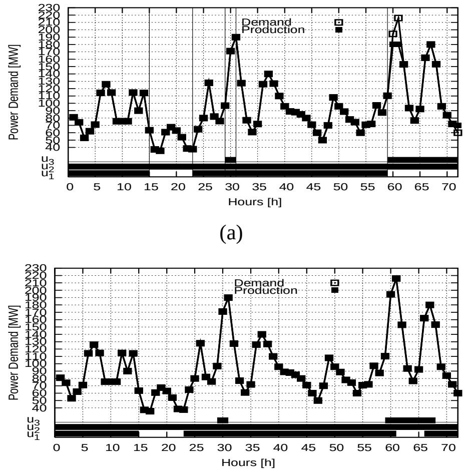
At the second level, three sub–intervals of 24 hours each were defined and solved by means of *EAs* with (if matching conditions were not satisfied) penalized objective function at their interfaces. The proposed method was able to find the global optimum (the one shown in Fig.4 (b)) after 18000 evaluations, each of which corresponds to a 24 hours (sub–interval) problem. So, this CPU cost is equal to approximately 4500 full problem evaluations. The algorithm converged after three cycles, each of which includes the sequential optimization of the three partitions. Fig.5 presents the objective function value of the optimal solution at the end of each intermediate cycle.

B. Case II

The second test case is concerned with the optimal schedule of a three unit system for a period of $T = 72$ hours and was handled using both one and two–objectives. The system features are given in Table III, whereas Fig. 6 shows the power demand distribution. The latter has been considered as fixed in the one–objective problem, or as the expected value in the two–objective one.

At the first level, the 72 hours were classified to five different operating scenarios, as shown in Table IV. Four sequential coarsening steps were carried out, shown in figs. 7(a) to 7(d), and ended up with a six group partition, shown in Fig.7(d). Therefore, the chromosome of the coarse–grained problem consists of only 18 digits. Fig.8(a) shows the optimal solution to the coarsened problem, computed by means of an *EA* whereas Fig.8(b) illustrates the global optimum computed at the second level. The latter was reached after 3800 evaluations, which corresponds to the 24–hour sub–problem, since on the second level three sub–intervals of 24 hours each were used.

The same problem was re–examined by considering that the power demand distribution is a stochastic one. A Gaussian



(b)

Fig. 8. Case II: Optimal solutions computed at (a) the first and (b) second optimization level, accordingly.

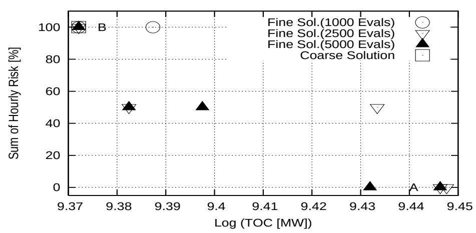
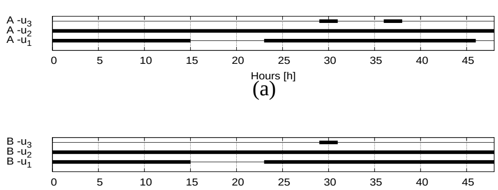


Fig. 9. Case II: Optimal solution fronts during the two–objective optimization method.



(b)

Fig. 10. Case II: Two extreme solutions selected from the optimal Pareto front solutions of Fig.9

distribution was assumed, with mean values the ones shown in Fig.6 and a variance of 1.5% of the mean value. The $SPEA2$ [9] method was used to compute the utility function within the same *EA* algorithm.

The non–dominated solutions at three instants of the evolution (after 1000, 2500 and 5000 evaluations) along with the Pareto front computed for the coarse problem are shown in Fig. 9. Figure 10(a) and 10(b) present the "extreme" solutions over the Pareto front (Fig.9), denoted as A and B respectively. Solution A is less risky but more costly than B. Notice that a possible demand surplus at the 37^{th} hour ($d^{(37)}=140MW$) cannot be handled out by the first two units, which may together produce $140\ MW$ at most. For this reason, in solution A the third unit turns *ON* at the 35^{th} hour. On the other hand, B "takes the risk" to keep u_3 *OFF*, despite the risk. Similarly, A turns *OFF* u_1 at the 47^{th} hour, because of the possible demand reduction (beneath $50\ MW$, which can be produced u_1 and u_2, operating at minimum load). Solution B takes the risk to keep both u_1 and u_2 *ON*, in favor of less fuel consumption, needed for shutting–down u_1.

V. Conclusions

An efficient two–level optimization algorithm for the solution of unit commitment problems has been presented and demonstrated on two test problems. The proposed method is based on evolutionary algorithms, coarsening heuristics and repairing schemes. The first level provides a good starting solution which is further refined at the second level. The latter is based on the partitioning of the time–interval and the iterative –sequential solution over each sub–interval, up to convergence. The proposed algorithm has also been used to solve two–objective optimization problems as well, by accounting for uncertainties in the power demand distribution.

Acknowledgment

This work is co–funded by the PENED03 program (Measure 8.3 of the Operational Program Competitiveness, of which 75 % is European Commission and 25 % national funding), under project number $03ED111$, and the Lavrio Power Production Center of the Greek Power Production Corporation S.A.

The authors would like to thank D. Pagoulatos, E. Bonataki and A. Kyrsanidi (Public Power Corporation S.A.) for their assistance regarding the formulation of the *UC* problem.

References

[1] N. P.Padhy, *Unit Commitment- A Bibliographical Survey*, IEEE Transactions on Power Systems,Vol. 19(2), May 2004.

[2] A. G. Bakirtzis, S. A. Kazarlis, V. Petridis, *A Genetic Algorithm Solution to the Unit Commitment Problem*, IEEE Trans. Power Systems, Vol. 11(1), 1996.

[3] A. G. Bakirtzis, C. E. Zournas *Lambda of Lagrangian Relaxation Solution to Unit Commitment Problem*, Generation, Transmission and Distribution, IEE Proceedings, Vol.147, March 2000.

[4] I. G. Damousis, A. G. Bakirtzis, P. S. Dokopoulos, *A Solution to the Unit Commitment Problem Using Integer Coded Genetic Algorithm* , IEEE Trans. Power Systems, Vol. 19(2), May 2004.

[5] A. Y. Saber, T. Senjyu, T. Miyagi,N. Urasaki, T. Funabashi, *Fuzzy Unit Commitment Scheduling using Absolutely Stochastic Simulated Annealing*, IEEE Trans. Power Systems, Vol. 21 (2), May 2006.

[6] D. Simopoulos, S. Kavatza,C. D.Vournas *Unit Commitment by an Enhanced Simulated Annealing Algorithm*, IEEE Trans. Power Systems, Vol. 21(1), Febr. 2006.

[7] M. K. Karakasis, D. G. Koubogiannis, K. C. Giannakoglou, *Hierarchical Distributed Evolutionary Algorithms in Shape Optimization*, Int. J. for Numerical Methods in Fluids, Vol. 53, pp.455-469, 2007.

[8] G. Ferrari-Trecate, E. Gallestey, P. Letizia, M. Spedicato, M. Morari, M. Antoine *Modeling and control of co-generation power plants: a hybrid system approach*, IEEE Trans. Control Systems Technology, Vol.(12), Sept. 2004.

[9] E. Zitzler, K. Deb and L. Thiele*Comparison of Multiobjective Evolutionary Algorithms:Empirical Results*, TIK-Report No. 70,Computer Engineering and Communication Networks Lab,ETH Zurich, 1999.

[10] J. Nocedal and S. J.Wright *Numerical Optimization*, Springer–Verlag New York, ISBN-0-387-98793-2, 1999.

[11] http://velos0.ltt.mech.ntua.gr/EASY