J. Ugirumurera and Z.J. Haas, "Power Scheduling for Programmable Appliances in Microgrids," 20th IEEE International Workshop on Computer Aided Modelling and Design of Communication Links and Networks, Special Session on ICT-Based Solutions for the Smart Energy Grid, September 7-9, 2015, University of Surrey, Guildford, UK

Power Scheduling for Programmable Appliances in Microgrids

Juliette Ugirumurera[1], Zygmunt J. Haas[1,2], *Fellow, IEEE*

[1]Department of Computer Science,
The University of Texas at Dallas
Richardson, Texas 75080
ugirumurera@utdallas.edu

[2]School of Electrical and Computer Engineering
Cornell University
Ithaca, NY 14850
zjh1@cornell.edu

***Abstract*- Recent advances in computing and communication enable the concept of smart homes that contain programmable appliances. Knowing that most household tasks do not need to be performed at specific times, but rather within a preferred time period, this paper studies the problem of optimal power generation scheduling in an isolated Microgrid, exploiting the flexibility to schedule energy-consuming tasks in smart homes. We formulate the problem as a non-linear optimization problem and present two scheduling protocols to solve it: GA-INT, a genetic algorithm that utilizes task interruptions, and PRO-S, a heuristic-based algorithm, which strives to smooth out peaks in the load profile. Numerical simulations demonstrate that PRO-S successfully reduces the complexity of the problem, while guaranteeing performance that approximates GA-INT's. The latter returns optimal or nearly optimal solutions, but with long execution times.**

Keywords—Microgrid; Programmable Appliances, Optimal Generation Scheduling; Unit Commitment Problem, Genetic Algorithm; Smart Homes;

I. INTRODUCTION

The optimal scheduling of power generation, also known as Unit Commitment (UC) problem, is one of the most challenging problems in power systems optimization [1]. In a Microgrid (MG), which is basically small scale power system with the ability to self-supply and islanding, the MG Central Controller (MGCC) has to coordinate the MG distributed generation (DG) sources in order to provide enough power to satisfy the load demand, while striving to achieve some optimal objective. This usually involves determining hundreds of discrete and continuous variables subject to numerous linear, quadratic, and sometimes non-linear constraints depending on the DG source characteristics and load demands. The DG sources can comprise of different technologies including Diesel engines, micro turbines, fuel cells as well as photovoltaics, wind turbines, and hydro turbines, with capacity varying from few kW to 1-2 MWs ([2], [3]).

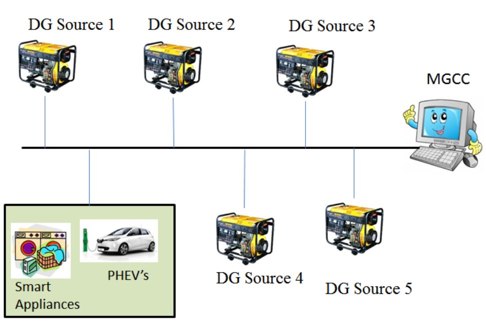

Fig. 1: System under Consideration

In this work, we address the UC problem in the context of an "islanded" MG that supplies electricity for smart homes. The smart homes contain programmable appliances, which can be scheduled for operation [4]. We consider appliance operations as schedulable tasks with power and timing demands. The smart appliances communicate with the MGCC about user-scheduled tasks, for example, over power lines. Fig. 1 depicts an exemplary schematic description of the system under consideration.

We assume that tasks get scheduled on a day-ahead basis, so that the MGCC can schedule the DG sources' operation for the upcoming day. We do not consider the power demands due to spontaneous small loads, such as TV set, computer, or microwave oven, but rather assume that the MG has reserved generator capacity to produce enough continuous power to meet those needs. Additionally, we only focus on scheduling dispatchable power sources, such as Diesel engines and fuel cells, and we leave the integration of renewable energy sources for future work.

II. RELATED WORKS

The UC problem in MG has gained interest in the past few years. Works [5]-[9] as well as [1] study MG optimal power generation for forecasted fixed (mostly hourly) electric loads. In contrast, references [10]-[14] investigate the UC problem for MG with schedulable load/tasks, which is similar to our work. However, unlike [10], [11], and [12], our work is based on a practical generators' model, which includes startup costs and generators' states in addition to energy-consuming tasks scheduling. Furthermore, we allow the optimization procedure to interrupt and resume tasks execution, which is in contrast to non-interruptible tasks in [10] and [12]. We also focus our work on scheduling the operation of thermal units, which is in contrast to Angelis et al. work in [13] that considers main grid energy and renewable energy. Though [14] considers the startup costs of dispatchable sources in the problem formulation, these costs are not used in the numerical simulations presented in that paper.

A genetic algorithm (GA), GA-INT, is used to solve the non-linear optimization problem formulated. GAs are search techniques based on the principal and mechanism of natural selection and "survival of the fittest" from natural evolution ([15]). GAs have been proven to be efficient in solving problems similar to UC problem, and, in recent decades, have been successfully applied to UC problems in power system ([9], [16]-[21]). Our contribution also includes a heuristic based algorithm, PRO-S, which seeks to flatten the load profile in order to reduce the extra costs due to DG Sources' on/off switching. PRO-S greatly decreases the computing time needed to solve the problem, while incurring only negligible cost penalties.

Thus, our work's contributions include:

- Formulating the UC problem for MG utilizing a more realistic generators' model, which comprises startup costs and generators states.
- Determining generators' optimal power scheduling, exploiting tasks scheduling, especially tasks pausing.
- Evaluating the effect of task schedulability and generators' startup costs on MG operation costs via simulations.
- Designing PRO-S, a heuristic algorithm that greatly reduces the problem's time complexity, while earning minimal cost increases.

III. System Model

We use a discrete time model, where the total scheduling time is T timeslots, which corresponds to 24 hours. Knowing that it takes few minutes to start up a thermal generator, a shorter sampling rate (5 min) is used in our model.

A. The DG Source Model

The MG consist of N uniform generating units, characterized by production cost coefficients ca and cb, where ca is the maintenance cost per timeslot (in \$/timeslot), and cb is the fuel cost per timeslot per kilowatt (in \$/kW). The generators have the same power generating capacity PG (in kW). Each generator also has a time-dependent startup cost $SC_n(t)$ (in \$/timeslot). In practice, a generator's capacity varies between a minimum and a maximum power generating limit ([22]), however, in this work, we assume constant output power for simplicity. We also assume that shutdown cost for each generator is equal to zero. The total cost to generate PG kW is found by [23] to be:

$$C(PG) = ca + cb * PG \tag{1}$$

The generators also have a minimum up time TU, and a minimum down time TD. The violation of such constraints can lead to shortness in the generating unit's lifetime ([5]). The startup cost $SC_n(t)$ depends on how long a generator has been off ([1]) by timeslot t:

$$SC_n(t) = \begin{cases} hc : TD \le toff_n(t) \le TD + TC \\ cc : toff_n(t) > TD + TC \end{cases} \tag{2}$$

In (2), $toff_n(t)$ is the continuous off time of unit n by time t, and TC is the cold start time for a generator. hc and cc are the host startup cost and cold startup cost, respectively.

Each source also has startup time ST (in timeslots), which is the necessary time to switch the generator from the off state to the active state. We also consider the generators initial states using G_n, and L_n. G_n is the number of timeslots generator n has to be initially on due to TU, while L_n is the number of timeslots source n has to be off at the outset due to TD.

B. The Task Model

We consider J tasks planned by customers for their appliances to be performed the next day. Each task j is characterized by the tuple $\{p_j, r_j, s_i, l_i\}$, where p_j is task j's power demand (in kW), r_j is its duration (in timeslots), s_i is its earliest possible start time, and l_i is its latest possible finish time.

C. The Task Allocation Model

Similar to the job-to-server allocation model used in [24], we design a J x T x N matrix A to keep track of tasks' allocation to the different generators during the considered time. In this matrix, an entry $a_{j,t,n}$ indicates the amount of power produced by generator n for the task j during the timeslot t. A generator cannot produce negative power, and the power generation limit for each generator has to be maintained:

$$a_{n,t,j} \ge 0 \tag{3}$$

$$\sum_{j=1}^{J} a_{j,t,n} \le PG \tag{4}$$

A horizontal plane matrix in A is an N x T dimensional matrix A_j that shows power generation for task j on the different generators over the T timeslots. Given a matrix A_j, we define a unary matrix operation $lz\colon A_j \rightarrow Z^+$, which returns the number of all-zeroes leading columns in A_j. Task j starts execution at $x_j = lz(A_j) + 1$. Power scheduling has to insure that x_j is greater or equal to s_j, the earliest start time of task j:

$$x_j \ge s_j \tag{5}$$

We design another function $tz\colon A_j \rightarrow Z^+$, which returns the number of all-zeroes trailing columns in A_j. We can then find task j's finish time as: $f_j = T - tz(A_j)$. f_j has to be less or equal to l_j in order to meet the task's deadline:

$$f_j \le l_j \tag{6}$$

We also use a unary function $nz\colon A_j \rightarrow Z^+$ to determine the number of non-zero columns (columns with at least one non-zero entry) in A_j, so as to find the number of timeslot where power was generated for task j. The number of non-zero columns for a task j has to be equal to its duration value.

$$nz(A_j) = r_j \tag{7}$$

$$x \in \{0,1\}, \sum_{n=1}^{N} a_{n,t,j} = p_j * x \tag{8}$$

Equation (8) states that the power generated for a task j during slot t is either zero or p_j.

D. The DG Source States

We construct another N x T dimensional matrix B, where each entry indicates the state of a generator n during timeslot t. We let 0, 1, and 2 refer to the active state, the off state, and the startup state, respectively. Matrix B is obtained from matrix A, since the generators have to be on whenever they are producing power for tasks. We define the above constraints as:

$$b_{n,t} \in \{0,1,2\} \tag{9}$$

$$\left(\sum_{j=1}^{J} a_{j,t,n} \right) * b_{n,t} = 0 \tag{10}$$

Equation (10) ensures that generator n is on whenever matrix A indicates that unit n is generating power.

We define the following unary matrix operations to determine a generator's state during a particular timeslot t:

- $As\colon b_{t,n} \rightarrow Z^+$, which returns 1 if $b_{t,n}$ is equal to 0, and returns 0 otherwise.

- *Os:* $b_{t,n} \rightarrow Z^+$, which returns 1 if $b_{t,n}$ has value 1, and returns 0 otherwise.
- *Ss:* $b_{t,n} \rightarrow Z^+$, which returns 1 if $b_{t,n}$ has value 2, and returns 0 otherwise.

We use the above operators to ensure that the generators' initial states are maintained as specified by G_n, and L_n:

$$\sum_{t=1}^{t+G_n-1} As(b_{n,t}) \geq G_n \tag{11}$$

$$\sum_{t=1}^{t+L_n-1} Os(b_{n,t}) \geq L_n \tag{12}$$

We also define allowable state transitions for the generators from one timeslot to the next, as shown in the Table I.

We use the following constraint to ensure generator allowable state change:

$$\forall n = 1,..., N, \forall t = 1,..., T,\ b_{n,t+1} \in \{b_{n,t}, (b_{n,t}+1) \bmod 3\} \tag{13}$$

The *mod* operator ensures that generators can switch from the startup state (indicated by 2) to the active state (designated by 0).

$$if\ b_{n,t} = 2\ and\ b_{n,t+1} = 0,\ \sum_{i=t+1}^{t+TU} b_{n,i} = 0 \tag{14}$$

$$if\ b_{n,t} = 1\ and\ b_{n,t+1} = 2,\ \sum_{i=t+1}^{t+TS} Ss(b_{n,i}) = TS \tag{15}$$

$$if\ b_{n,t} = 1\ and\ b_{n,t+1} = 2,\ b_{n,t+TS+1} = 0 \tag{16}$$

$$if\ b_{n,t} = 0\ and\ b_{n,t+1} = 1,\ \sum_{i=t+1}^{t+TD} Os(b_{n,i}) = TD \tag{17}$$

Constraint (14) indicates that a generator that switches from the startup state to the active state has to stay on for at least *TU* timeslots, while constraint (15) specifies that a generator *n* that switches from the off state to the startup state has to spend *TS* timeslots in the startup state. Constraint (16) states that after *TS* startup timeslots, the generator in the startup state has to be on. Constraint (17) indicates that a generator in the active state that is shut down will remain off until at least *TD* timeslots have elapsed.

E. Problem Statement

Our goal is to determine the generating units' states during the *T* timeslots, so as to minimize the total operating costs, while meeting the tasks' power and timing requirements. The MG operation cost, *C*, is calculated from the generators' power production cost and the startup costs. We state the optimization problem as follows:

$$\begin{aligned} Min \quad C = & \sum_{t=1}^{T}\sum_{n=1}^{N} As(b_{n,t}) * (ca + cb * PG) + \\ & \sum_{t=t_1}^{t_f}\sum_{n=1}^{N} Ss(b_{t,n}) * SC_n(t) \end{aligned} \tag{18}$$

such that (3)-(17) hold.

IV. Solution Methods

A. Genetic Algorithm:

The problem as formulated above is a non-linear mixed integer programing problem. We design a genetic algorithm, GA-INT, to solve it. We implement GA-INT in Matlab on a 3.20 GHz Intel Core computer with 4GB of RAM. The parameters in the GA-INT are described in Table II.

TABLE I: Allowable State Transitions

State at *t*	**State at *t+1***
0	0, 1
1	1, 2
2	2, 0

B. Heuristic Algorithm

The heuristic algorithm, PRO-S, is also implemented in Matlab. As shown in Fig. 2, PRO-S first creates a *J* x *T* matrix, *D*, and populate it in the following way: each row, D_j, corresponding to task *j*'s power consumption, is populated with value p_j in columns $D[j,a_j]$ through $D[j,l_j]$. The remaining entries in D_j contain zeros. Each row D_j contains $l_j-a_j+1-r_j$ extra p_j values. We derive a load profile array *Q* of length *T*, such that:

$$Q[t] = \sum_{j=1}^{J} D[j,t]. \tag{19}$$

In order to remove the extraneous p_j, PRO-S proceeds in a greedy manner. In each iteration, PRO-S identifies elements in *Q* with the largest value q_{max}. It then tries to lower q_{max} in *Q* by zeroing out some extraneous p_j's in *D* that contribute to the q_{max} entries, starting with those rows with the largest p_j values. At the end of the iteration, PRO-S updates *Q* from the current *D* matrix. It also saves the current q_{max} value, so that in the following iterations only those *Q* entries with value less than q_{max} are considered. PRO-S repeats this greedy choice until all extra p_j's are eliminated from *D*.

The goal of PRO-S is to finish with a load curve that is as smooth as possible, which minimizes the change in power production from one slot to the next. We use the final load array *Q* to determine, u_t, the required number of active generators in each slot *t* by:

$$u_t = ceil\left(\frac{Q[t]}{PG}\right) \tag{20}$$

Using the u_t values, we create a generator state matrix *B,* so that each timeslot *t* has u_t active generators. For each *t*, the active generators are chosen starting with those generators with positive G_n values, since these generators are already on by the beginning of the scheduling period. The remainder of the generator states is determined so as to minimize the additional costs; i.e., a generator is turned off whenever it is not needed, unless incurring the startup cost is more expensive, in which case it is kept on. From matrix *B*, we determine the total operating cost using (18).

V. Case Study

We simulated an isolated MG powering a remote neighborhood of smart homes. The MG is made up of small identical thermal units and their characteristics are shown in Table III. Each home is assumed to contain at most seven programmable appliances, where each appliance submits daily a number of tasks. The tasks' arrivals were generated following a Poisson distribution, with the constraint that the l_j-s_j could not be more than $4*r_j$. Other tasks' characteristics are shown in Table IV, where M_k is the daily arrival rate per task type, and H_k refers to the average number of tasks per task type, per home. We compare PRO-S to GA-INT, and show that PRO-S' performance is nearly as good as GA-INT's.

TABLE II: GA Parameters

Parameter	Value
Population Size	1000
Probability of Crossover	60%
Probability of Mutation	40%
Stopping Criteria	No Improvement in fitness value for 50 generations

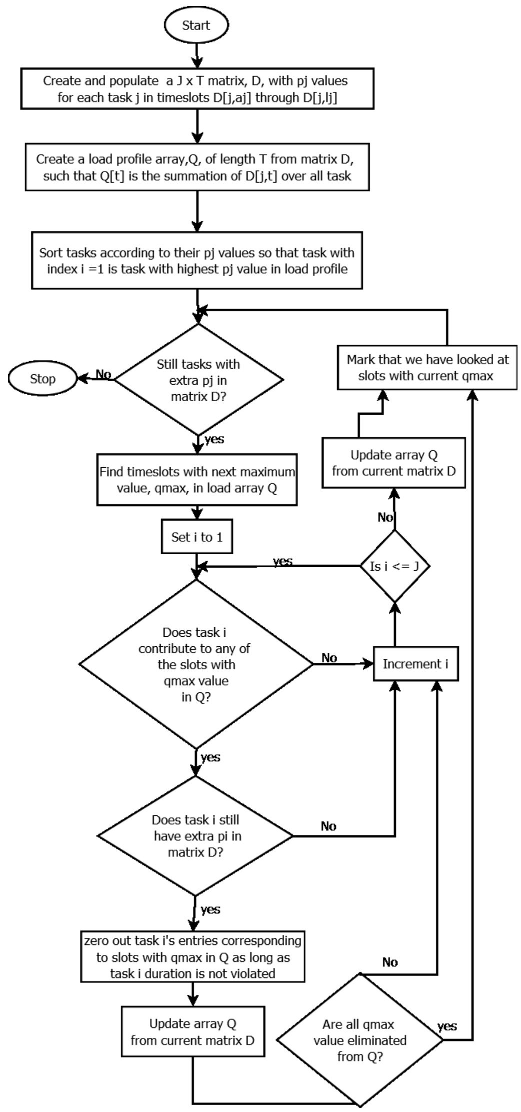

Fig. 2: Heuristic Algorithm, PRO-S

TABLE III: Parameter of DG Sources

Generator Characteristics	
PG (kW)	2.12
ca ($/timeslot)	0.015
cb ($/kW)	0.005
hc ($/timeslot)	0.2
cc ($/timeslot)	0.4
TS (timeslots)	2
TU (timeslots)	5
TD (timeslots)	2
TC (timeslots)	2

TABLE IV: Electricity Consuming Tasks [25]

Task Type	p_j (kW)	r_j (timeslots)	M_k (tasks/day)	H_k (tasks/home)
Space Heater	3.4	18	60	6
Electric Car	3.5	30	30	3
Spin Dryer	3	12	30	3
Air Conditioner	3	12	80	8
Laundry Machine	1.5	6	20	2
Swimming Pool Heating	4.5	24	10	2
Dish Washer	1	8	20	2

The performance of GA-INT and PRO-S are both compared to the Early Starting Time (EST) scheme ([10]). In the EST scheme, domestic appliances are turned on at their given earliest starting time, which is similar to common living habits where users turn on appliances as soon as they want to use them.

A. Test Case Results

We first present results from one simulation trial, where appliances submit a total of 250 tasks, with a mean load of 42.3681 kW. As depicted in Fig. 3, without load scheduling, this simulation instance requires at least 39 generators in order to meet the peak load. On the other hand, GA-INT requires only 24 generators, while PRO-S needs no more than 26. Fig. 4 demonstrates that PRO-S' load scheduling closely follows GA-INT's. Hence, in this case, GA-INT's cost reduction over PRO-S was less than 1% (Table V).

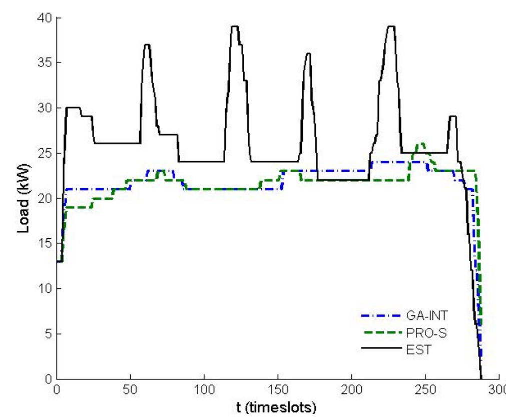

Fig. 3: Comparison of Number of Active Generators vs. time

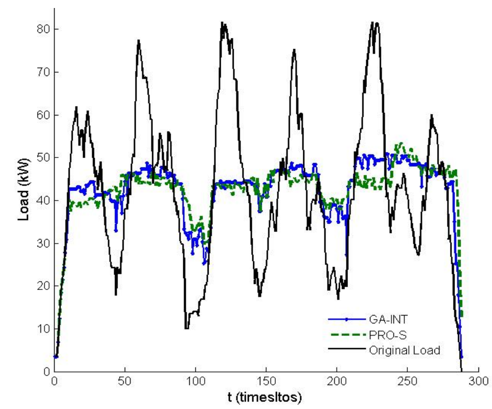

Fig. 4: Load Scheduling Comparison

TABLE V: Cost Reduction Results

Algorithm	Operation Cost ($)	Cost Reduction over EST (%)	GA-INT Cost Reduction over PRO-S (%)
GA-INT	167.7632	31.52	0.95
Heuristic	169.376	30.86	
ETS	244.976		

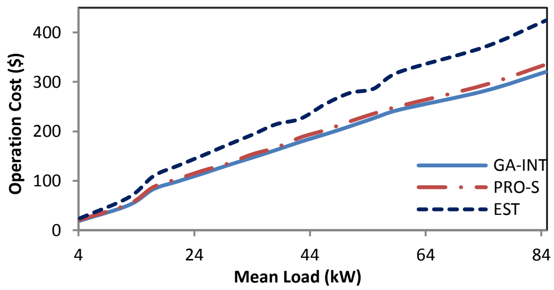

Fig. 5: Operation Costs Comparison

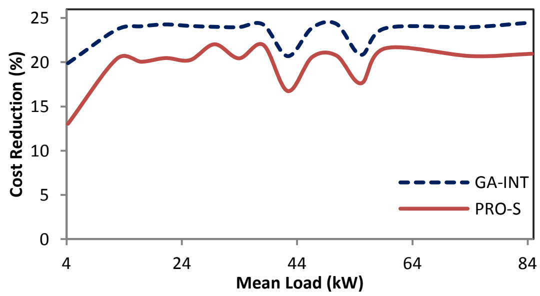

Fig. 6: Cost Reduction over EST

TABLE VI: GA-INT and PRO-S Time Comparison

Mean Load (kW)	GA-INT (s)	PRO-S (s)
4.24	4821	10.56
16.95	21189	8.24
25.42	15753	8.81
33.89	27633	7.18
42.37	35653	7.10
46.60	32259	7.64
55.08	33110	8.37
59.32	67144	7.59
74.14	49953	7.37
84.74	63068	6.69

B. Varying Mean Load

We compare PRO-S to GA-INT and EST in scenarios where the mean load varies. In each test scenario, the tasks' p_j values are multiplied by a constant that changes from 0.1 to 2, which in turn changes the mean load by the same factor. The generators' capacity is kept the same. For each scenario, we run 10 instances that have the same mean load, but differ in their tasks starting times and deadlines values as well as generators initial conditions. We then determine the average cost reductions observed over those instances.

Fig. 5 shows that PRO-S operation costs follow closely GA-INT's as the mean load increases, while Fig. 6 demonstrates that GA-INT's registers no more than 7% cost savings over PRO-S, which falls below 5% as the load increases. This emphasizes that PRO-S performance is nearly as good as GA-INT's whenever the load demand is not negligible. An important advantage of PRO-S is that PRO-S greatly reduces the problem's time complexity, solving it in few seconds only, while GA-INT needs about 4800 sec for the same case, which increases with the mean load (Table VI).

C. Varying the Number of Tasks

We also evaluate how PRO-S performs vis-à-vis GA-INT and EST as the tasks' daily arrival rate increases from 50 up to 500 tasks. A 50 tasks system corresponds to a two-home model, while 500 tasks simulate a 20 home system. For each model, we run 10 trials with the same number of tasks, and average out the results. As before, the 10 trials deviates in their tasks' starting times and deadlines values, and generators initial conditions.

From Fig. 7, we notice that PRO-S' operation costs approximate GA-INT's even as the number of submitted tasks grows. Fig. 8 shows that GA-INT performance improvement over PRO-S diminishes from about 9% to less than 5% as the tasks' arrival rate rises. Fig. 8 also illustrates that GA-INT and PRO-S cost reduction over EST decreases as the number of submitted tasks rises. This is due to the system's load demand becoming more constant as the task arrival rate grows. This reduces the opportunity for load scheduling to curtail costs.

We also observe that GA-INT's and PRO-S' curves in Fig. 8 are not smooth. This is explained by generators having a constant output power, so that even if the load demand requires only 0.1 kW from a generator, the generator still outputs 2.12 kW. Fig. 9 shows PRO-S' cost reduction curve when the tasks' p_j values are multiplied by 21.2, while the generators capacity remains the same. This curve is more regular compared to PRO-S' curve when the p_j values are unchanged. Thus, as the p_j's grow, the generators' output power becomes smaller and more continuous in relation to the tasks' load demand, which smoothens out the cost reduction curve. GA-INT's cost reduction profile also evens out as p_j's rise, since PRO-S performance nears GA-INT's.

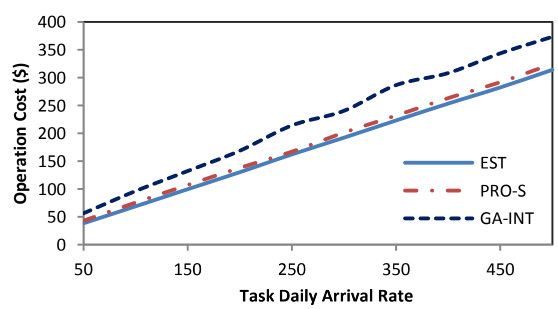

Fig. 7: Operation Costs Comparison

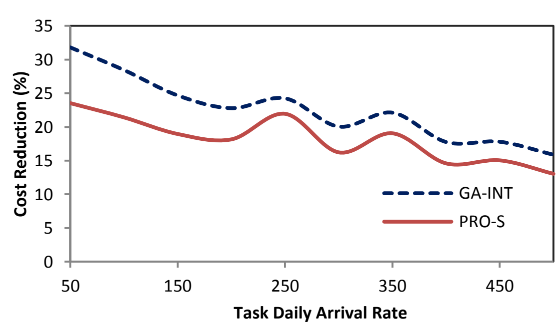

Fig. 8: Cost Reduction over EST

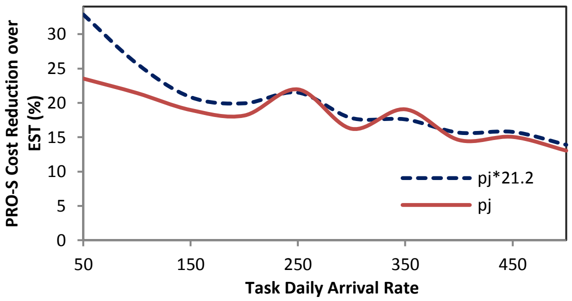

Fig. 9: PRO-S Cost Reduction with p_j*21.2 vs PROS Cost Reduction with p_j

Table VII: Time Complexity Comparison

Daily Task Arrival Rate	GA-INT (s)	PRO-S (s)
50	3598	0.60
100	7825	1.13
150	11675	2.75
200	36112	7.42
250	39720	6.597
300	54141	10.43
350	49717	8.36
400	59563	12.00
450	84111	12.51
500	96306	23.00

In this case again, PRO-S only needs few seconds on average to solve the problem, while GA-INT requires at least an hour to solve a 50 task problem and more than 10 hours when the number of daily tasks is greater than 200 (Table VII).

Hence, since we know that GA-INT finds the best or nearly the best solution, we conclude that PRO-S returns solutions that are also close to the optimal solution, especially when the mean load and task arrival rate are high.

VI. Conclusion

This paper formulates the problem of optimal power generation in an "isolated" MG as a non-linear mixed integer problem, and implements two algorithms, GA-INT and PRO-S, to solve it. GA-INT is a genetic algorithm that exploits task interruption and task shifting, and produces optimal or near-optimal solutions. Since GA-INT is time expensive, we also design a heuristic-based algorithm, PRO-S, to reduce the complexity of the problem. Simulation results demonstrates that, in medium to high load situations, PRO-S indeed reduces greatly the time complexity of the problem by solving it in few seconds, while incurring less than 5% in extra cost in comparison to GA-INT. The latter requires at least an hour in low load situations, and can take more than 10 hours when the daily task arrival rate is greater than 200. However, even in low load scenarios, GA-INT cost reduction was no more than 9% over PRO-S.

Acknowledgment

This work was supported in part by a grant from the U.S. National Science Foundation, grant number ECSS-1308208.

References

[1] T. Logenthiran and D. Srinivasan, "Short term generation scheduling of a Microgrid," in *TENCON 2009 - IEEE Region 10 Conf.*, 2009, pp. 1–6.

[2] "MICROGRIDS - Large Scale Integration of Micro-Generation to Low Voltage Grids," *EU Contract ENK5-CT-2002-00610, Technical Annex*. [Online]. Available: http://microgrids.eu/micro2000/presentations/19.pdf. [Accessed: 24-Sep-2014].

[3] A. G. Tsikalakis and N. D. Hatziargyriou, "Centralized control for optimizing microgrids operation," in *2011 IEEE Power and Energy Society General Meeting*, 2011, pp. 1–8.

[4] S. U. Z. Khan, T. H. Shovon, J. Shawon, A. S. Zaman, and S. Sabyasachi, "Smart box: A TV remote controller based programmable home appliance manager," in *2013 International Conference on Informatics, Electronics and Vision (ICIEV)*, 2013, pp. 1–5.

[5] F. A. Mohamed and H. N. Koivo, "System modelling and online optimal management of MicroGrid using Mesh Adaptive Direct Search," *Int. J. Electr. Power Energy Syst.*, vol. 32, no. 5, pp. 398–407, Jun. 2010.

[6] H. Morais, P. Kádár, P. Faria, Z. A. Vale, and H. M. Khodr, "Optimal scheduling of a renewable micro-grid in an isolated load area using mixed-integer linear programming," *Renew. Energy*, vol. 35, no. 1, pp. 151–156, Jan. 2010.

[7] R. Palma-Behnke, C. Benavides, F. Lanas, B. Severino, L. Reyes, J. Llanos, and D. Saez, "A Microgrid Energy Management System Based on the Rolling Horizon Strategy," *IEEE Trans. Smart Grid*, vol. 4, no. 2, pp. 996–1006, Jun. 2013.

[8] A. Parisio, E. Rikos, and L. Glielmo, "A Model Predictive Control Approach to Microgrid Operation Optimization," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 5, pp. 1813–1827, Sep. 2014.

[9] H. Z. Liang and H. B. Gooi, "Unit commitment in microgrids by improved genetic algorithm," in *2010 Conference Proceedings IPEC*, 2010, pp. 842–847.

[10] D. Zhang, N. Shah, and L. G. Papageorgiou, "Efficient energy consumption and operation management in a smart building with microgrid," *Energy Convers. Manag.*, vol. 74, pp. 209–222, Oct. 2013.

[11] Y. Zhang, N. Gatsis, and G. B. Giannakis, "Robust Energy Management for Microgrids With High-Penetration Renewables," *IEEE Trans. Sustain. Energy*, vol. 4, no. 4, pp. 944–953, Oct. 2013.

[12] D. Zhang, S. Liu, and L. G. Papageorgiou, "Fair cost distribution among smart homes with microgrid," *Energy Convers. Manag.*, vol. 80, pp. 498–508, Apr. 2014.

[13] F. De Angelis, M. Boaro, D. Fuselli, S. Squartini, F. Piazza, and Q. Wei, "Optimal Home Energy Management Under Dynamic Electrical and Thermal Constraints," *IEEE Trans. Ind. Informatics*, vol. 9, no. 3, pp. 1518–1527, Aug. 2013.

[14] A. Khodaei, "Microgrid Optimal Scheduling With Multi-Period Islanding Constraints," *IEEE Trans. Power Syst.*, vol. 29, no. 3, pp. 1383–1392, May 2014.

[15] D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning," Oct. 1989.

[16] S. Jalilzadeh and Y. Pirhayati, "An Improved Genetic Algorithm for unit commitment problem with lowest cost," in *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, 2009, vol. 1, pp. 571–575.

[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] T. T. Maifeld and G. B. Sheble, "Genetic-based unit commitment algorithm," *IEEE Trans. Power Syst.*, vol. 11, no. 3, pp. 1359–1370, 1996.

[19] A. H. Mantawy, Y. L. Abdel-Magid, and S. Z. Selim, "Integrating genetic algorithms, tabu search, and simulated annealing for the unit commitment problem," *IEEE Trans. Power Syst.*, vol. 14, no. 3, pp. 829–836, 1999.

[20] K. S. Swarup and S. Yamashiro, "Unit commitment solution methodology using genetic algorithm," *IEEE Trans. Power Syst.*, vol. 17, no. 1, pp. 87–91, 2002.

[21] S. Orero, "Large scale unit commitment using a hybrid genetic algorithm," *Int. J. Electr. Power Energy Syst.*, vol. 19, no. 1, pp. 45–55, Jan. 1997.

[22] T. Niknam, A. Khodaei, and F. Fallahi, "A new decomposition approach for the thermal unit commitment problem," *Appl. Energy*, vol. 86, no. 9, pp. 1667–1674, Sep. 2009.

[23] S. X. Chen, H. B. Gooi, and M. Q. Wang, "Sizing of Energy Storage for Microgrids," *IEEE Trans. Smart Grid*, vol. 3, no. 1, pp. 142–151, Mar. 2012.

[24] T. Mukherjee, A. Banerjee, G. Varsamopoulos, S. K. S. Gupta, and S. Rungta, "Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers," *Comput. Networks*, vol. 53, no. 17, pp. 2888–2904, Dec. 2009.

[25] Electropaedia, "Electricity Demand." [Online]. Available: http://www.mpoweruk.com/electricity_demand.htm. [Accessed: 03-Oct-2014].