INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

A FRAMEWORK FOR JOINT SCHEDULING OF TASK AND BATTERY, FOR MAXIMIZING BATTERY LIFETIME IN REAL-TIME SYSTEMS

Mahta aali [1,2], Shahram jamali[3]

[1] Department Of Computer, Ardabil Science And Research Branch, Islamic Azad University, Ardabil, Iran
[2] Department Of Computer, Ardabil Branch, Islamic Azad University, Ardabil, Iran
[3] Department Of Computer Engineering, University Of Mohaghegh Ardabili, Ardabil, Iran

Abstract: -Many portable devices rely on batteries for their power supply. The capacity of the batteries is finite, and the duration with which one can use the device is limited by the battery lifetime. Accordingly, to increase the efficiency of these systems, energy consumption and also managing the use of the batteries are too important. Given the characteristics of the nonlinear behaviour of the battery, for maximizing battery life, which is related to the discharge pattern of batteries, is one of np-hard problems. This paper to extending the system lifetime and maximizing the efficiency of the battery, presents a greedy algorithm for dynamic voltage scaling according to battery and power consumption characteristics of the tasks. These tasks have deadline and should be done on the specific time. In order to test the proposed algorithm offered in this paper, we test it with three algorithms to compare the results. Simulation results show that the proposed method (gjtbs) in different conditions (with different workload of the system) maximized systems lifetime.

Keywords*:* Embedded, Real-Time Systems, Dvs, Battery-Aware Scheduling.

1. Introduction

Nowadays the use of portable equipment such as laptops is growing rapidly. These systems to meet their energy, they use batteries which have a limited capacity, and on the other hand, some of them are real-time systems which they required to run at a certain time. Accordingly, it is necessary to design algorithms that manage the energy consumption of the batteries, with considering the deadline in the implementation of real-time systems.

We know that the capacity of the battery is limited and depend on the battery lifetime [1]. Lifetime is the time of one discharge period of the battery, from full to empty. In most technologies of battery, if the voltage of the battery decreases less than the required system voltage (Cut-off Voltage), it can no longer supply the System requirements and the system cannot continue to work. In this case the energy stored in the battery is not necessarily finished, but it is not enough to meet the voltage of the battery, while we can use this battery for the system with less than this cut-off voltage. Although the battery lifetime depends mostly on its capacity and the level of the load applied to it, another important influence is how the battery is used, i.e., its usage pattern [3].

When a battery is continuously discharged, a high current will cause it to provide less energy until the end of its lifetime than a lower current. This effect is termed the rate-capacity effect. On the other hand, during

Vol.3 Issue 12, Pg.: 5-15
December 2015

periods of low or no discharge current, the battery can recover to certain extend? This is termed the recovery effect. These two effects cause the battery to have a non-linear behaviour, and make it as np-hard problems.
The battery model we use is the Kinetic Battery Model (kibam) of Manwell and mcgowan [7–9]. This model is very intuitive, and the simplest model that includes the two important non-linear battery properties, the rate-capacity effect and the recovery effect [6].
In the model the battery charge is distributed over two wells: the available-charge well and the bound-charge well. Charge flows from bound-charge well to available-charge well. The rate of charges that flows between this two well depends on the height of the wells.
According to the kibam model, in theory the best way to discharge the batteries in a multiple battery system is by using them in parallel. For a system with N identical batteries discharged with a continuous current I, The lifetime is then given by:

$$Tp,n= \frac{CN}{I} - \frac{1}{k'}\left(\frac{1-c}{c} - W\left(\frac{1-c}{c} e^{-\frac{-Nck'}{I}+\frac{1-c}{c}}\right)\right) \quad (1)$$

The system lifetime when using N batteries sequentially will be Nts, hence the maximum possible gain with N batteries GN is given:

$$GN=\frac{tp,N}{Nts} \quad (2)$$

One approach is to extend the battery lifetime is connecting one or more batteries to each other or using intelligent batteries with more cells instead of using a large and single-cell battery, due to the impact of recovery effect [5].
That manages the charge and discharge of the battery to ensure the battery lifetime. This leads to reliability of the battery. Smart batteries are of a new design of the multi-cell batteries. They use the cells properties to increase battery lifetime and ensure of safety for better performance. The main character of cells is self managing.
These batteries are consisting of three basic components:

- Cell: that stores voltage and Specifies current and voltage in each cell by using recovery effect.
- Cell switching circuit: connecting cells to each other to make a bigger voltage.
- Battery Cell Array Manager: The IBCA manager is the core of the intellbatt system. It provides the following functionalities: monitors cell status, schedules cells for load current delivery, and ensures the safety of the IBCA.

The SMBUS (System Management Bus) interface7 between intellbatt and the system will provide for information exchange between the two entities. The CSC (CELL SWITCHING CIRCUIT) connects cells to deliver the required current to the target system in which intellbatt is installed. CSC configuration is performed by the IBCA manager with a code word. Figure 1 illustrates the envisioned system.

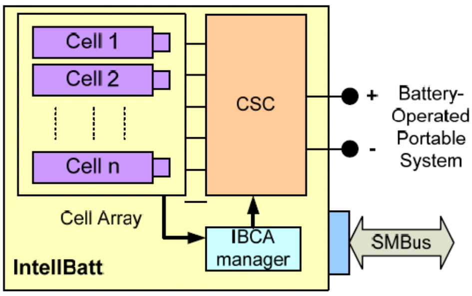


FIG 1: Intelligent battery Structure

The execution of works, named job scheduling, is another approach to increase battery life. In real-time systems deadline refers to the time that works must finished until that time, Otherwise they run low value, worthless, or even would be disastrous.

Response time, number of switching between battery cells and violations is the most important evaluation criteria. It should be noted, the response time is the time that a states on the system to run until it finished the work and run out of the system.

The rest of the paper is structured as follows. In Section 2 an overview is given of the related work. Section 3 describes the details of the proposed algorithm.. The results of the simulations are given in Section 4. Finally, we conclude and expressed ideas for future in the last Section.

2. Related works

In this section, the Scheduling algorithms presented in three sections named task scheduling, battery scheduling and joint battery and task Scheduling.

2.1 task scheduling algorithms

In this article [12] with the aim of optimizing power consumption to extend the systems lifetime, presented an algorithm which has 2 Phase with Analytical battery. The first phase guess the execution time of jobs in offline mode and the second phase used the slack distribution for scheduling tasks with same priority in online mode.

In 2006, paper [3] with the aim of increasing the battery lifetime suggests an algorithm for just task scheduling, which is consists of two steps. First, it specific the sequence of tasks and plotted them on the graph, and in the second step allocates the voltage to 2 model named static and dynamic voltage allocation. In the Static allocation, it allocates tasks to the specific battery and we cannot change the allocation, but in the dynamic allocation, it can allocate the tasks to batteries frequently.

In the proposed algorithm 8, prioritize tasks and enter them into the graph. The task with the highest priority is picked up and allocated to the battery. However, it should be noted that the graph has direction so they should not have loop.

Writers of [4] for optimizing power consumption in portable devices presented Battery Aware Compound Task Scheduling (BACTS). In this algorithm two steps Is offered: The first step is entering the tasks with regarding a threshold and in the second step it scales tasks voltage to run it.

In 2010 [14] the algorithm uses two battery properties and stated that in the event that a battery is activated, the other batteries are idle and they have time to recover the battery.

2.2 Battery scheduling algorithms

Mr. Benigni and Kstlyz article [20] In order to extend the system lifetime, concluded that battery scheduling is much better than when the batteries are connected in succession, Because in battery scheduling we use rate capacity and charge recovery. In this paper they compare the three algorithms. In the first case model describes the sequential battery scheduling that switching occurs only when it understand the battery is empty. The second case is Round-Robin scheduling static algorithm) in which the switching occurs at certain between the batteries and the third mode is the dynamic scheduling that switching can occur at any time. Recognize the right time for switching is crucial in this case.

In [7] two ways for maximizing battery lifetime with the pattern of battery usage in the real-time systems and the experimental simulation is presented. The first model is an analysis of the kinetic battery model, that scheduling battery uncertainly, in which that the battery can switches on the arbitrary points of the time. The second model is a discrete version of the kinetic battery model using a linear automate white taking into account the cost of the time. The decision for switching between batteries in predefined points of time is certain.

In 2010,[23] algorithm increase battery lifetime defined an algorithm, which consists of a matrix of Boolean. The name of this algorithm is BATA that is Abbreviation of Battery Aware Task Allocation which consists of several batteries. In this algorithm, the first task that comes into the system allocates to the best battery which is equal or more than the capacity of the task.

Authors of [33], in 2011, allocated tasks to batteries are mapped to a tree. The proposed algorithm is defined system as a discrete and variable. The algorithm is greedy to finding the sequential of tasks, which can do the most works as short as is possible and therefore less energy will be wasted.

Writers of [10] defined task scheduling with smart batteries. At first, the state of all cells was zero. After allocation, the number of these cells is altered to that work.

To run any work the algorithm sort voltages in ascending mode, then start to work according to the voltage level of works. If the works voltage was lower than cell voltage, it will allocate to that cell as a new bank. But

if the amount of works required voltage was more than the cells voltage, it will make a bigger bank of cells to execute the work by adding more cells.
This is done until the required voltage becomes less than or equal to the voltage of that task. In this case the batteries specified, connected sequentially, by cell switching circuit to do task in that bank. Authors on the results of the simulation conclude that whatever the number of these cells increased, it will be better. But again, it is unclear that how many of these cells will be the best.

2.3 Combination Scheduling

A. sequential scheduling

Type of scheduling, tasks allocated to cells or batteries as they arrived to system. In this case, the first task allocated to the first battery cell. If the task voltage was lower than the voltage of the cell, switches to the other cell. The next task will allocate to same battery until the battery finish or tasks complete.

B. Round-Robin scheduling (static allocation)

In this algorithm the first work allocated to the first cell or battery until it meets time quantum, then switching to the second cell battery, and this process will continue until the completion of the works or batteries.

C. best cell selection algorithm

In this algorithm, the desired task voltage search (greedy or etc) through cells and selects the best cells. Here, the best cell has the greatest voltage. It means that the first task with closer deadline selects the cell with largest voltage.

3. GJTBS algorithm

In this section we introduce the proposed algorithm with the name of greedy joint task and battery scheduling (GJTBS). To intended joint task and battery scheduling in intelligent batteries, we min intelligent batteries, must consider two important cases. First, specify the sequence of tasks and then allocate tasks to cells.
In the first phase, the main problem is specifying the sequence of tasks. As we know that our tasks are periodic, so the algorithm has a variable named T, Specifies during of the period. It is assumed equal to the length of the longest running time. Here the runtime time is equal to best runtime that is Approximate through statistical methods such as linear regression and etc was pre-planned and specified.
In this algorithm, as our system is real-time, tasks have deadline named D which is lifetime of the task and is should execute until that time, otherwise the execution of task is worthless.
In this paper, one method for evaluating the proposed algorithm is tasks response time. So we need to know the time of arrival and exit of tasks that write them to the array of tasks. In table 1 tasks specification has been shown.
In the second phase, the main problem is battery allocation. As the battery is assumed to be intelligent, so it formed of K cells. In this case, we have a constant voltage for the total battery named V. An array of voltages named vk. The values in this array representing the tasks voltage with the number K. The number of these cells is fixed, but its value can be constant or variable. Battery has capacity C, in which a fraction of this capacity is assigned to each cell. The difference value is due to the different voltage in the cells. Analytical battery model named Kinetic were used. Table 2 shows the intelligent batteries properties.
The allocation of tasks to the battery should be prospective. Scheduling without prospective means that, it may tasks arbitrary or without scheduling allocated to cells, it may Fails to provide the required voltage to execute the task or need to switch between cells. The Number of switching between cells is one of the most important evaluation criteria, whatever the number decrease, the computational overhead will decrease and makes the algorithm better and more efficient.

Variable name	Deadline	Execution time	Current	Voltage	Arrival time
Unit	Second	Second	Amp	Volt	Second

Table.1 Specification of the tasks

Variable	Cell voltage	Cell current	Cell capacity
Unit	Volt	Amp	A-H

Table.2 Specification of the intelligent batteries

Now we are going to explain the suggested algorithm (as shown in Figure2). In this algorithm, we have three different scenarios: the first scenario happens if the task required voltage is less than the sum of the all battery cells. The second scenario is the task required voltage is more than the sum of the all battery cells. And finally, the third scenario happens when the task require voltage is equal to the sum of the all battery cells.

First of all, this algorithm compares the voltage of the task with the cells voltage, if first and second scenarios occur, to achieve near-optimal solution GJTBS follows 4 steps.

First step: sequence of tasks

At this point, algorithm sorts tasks based on the nearest deadline. .In this case, task with a highest priority has a short deadline, which should execute faster.

Second step: total allocation

This step is the most basic step in the GJTBS algorithm. In this step, the algorithm greedy searches the battery cells and looking for the best place to executes task. For this purpose, it compares the cell voltage with the voltage of the task. Here, the best cell is the cell with the lowest voltage difference. This difference is greater or equal to zero. During the allocation we should attend to tasks deadlines. For this reason, the proposed algorithm before the final allocation should check that is the task meets the deadline or not? If the answer was negative, we allocate it and go on to the next task. But if positive, it must add one unit to violation number and record the number of task to the array of violation. The aim is completing all or most of the tasks be for their deadline.

Here, the array of allocation assigns with 0(not allocated yet), -1(miss deadline) and positive numbers (the number of task that allocated to this cell).

Step three: assign tasks to cells by switching.

In this step, those tasks that have not been able to execute in the previous step, are going to be allocated to cells. In this step, the algorithm Looking for the best cell from the beginning of allocating array. To this purpose, GJBTS greedy searches between the cells with zero value of allocation array to gather them.

To this end, we re-examine whether task meet deadline or not. If this happened, it changes the number to -1 and increase one unit of the violation number. If not, it sort cells voltage in descending mode, then from the beginning of the array starts to gathering voltages until tasks voltage become less than or equal to the cells voltage. As switching between cells is important, during the aggregating voltage, we used switching counter for each task.

Step four: rest batteries

At the end of each period, based on charge recover of kinetic batteries (as was presented in the first part), it added the amount of voltage which recovers at the rest time.

If the third scenario occurred, as we cannot execute all tasks, so our aim is to completing more tasks in the battery lifetime. For this purpose, in the first step, we sort tasks voltage in ascending, and cells voltage in descending. In the second step, algorithm fully allocates cells to tasks. For assign tasks to cells by switching, GJTBS looks greedy for gathering available voltages to execute task. In Figure 2, flowchart of GJTBS is shown.

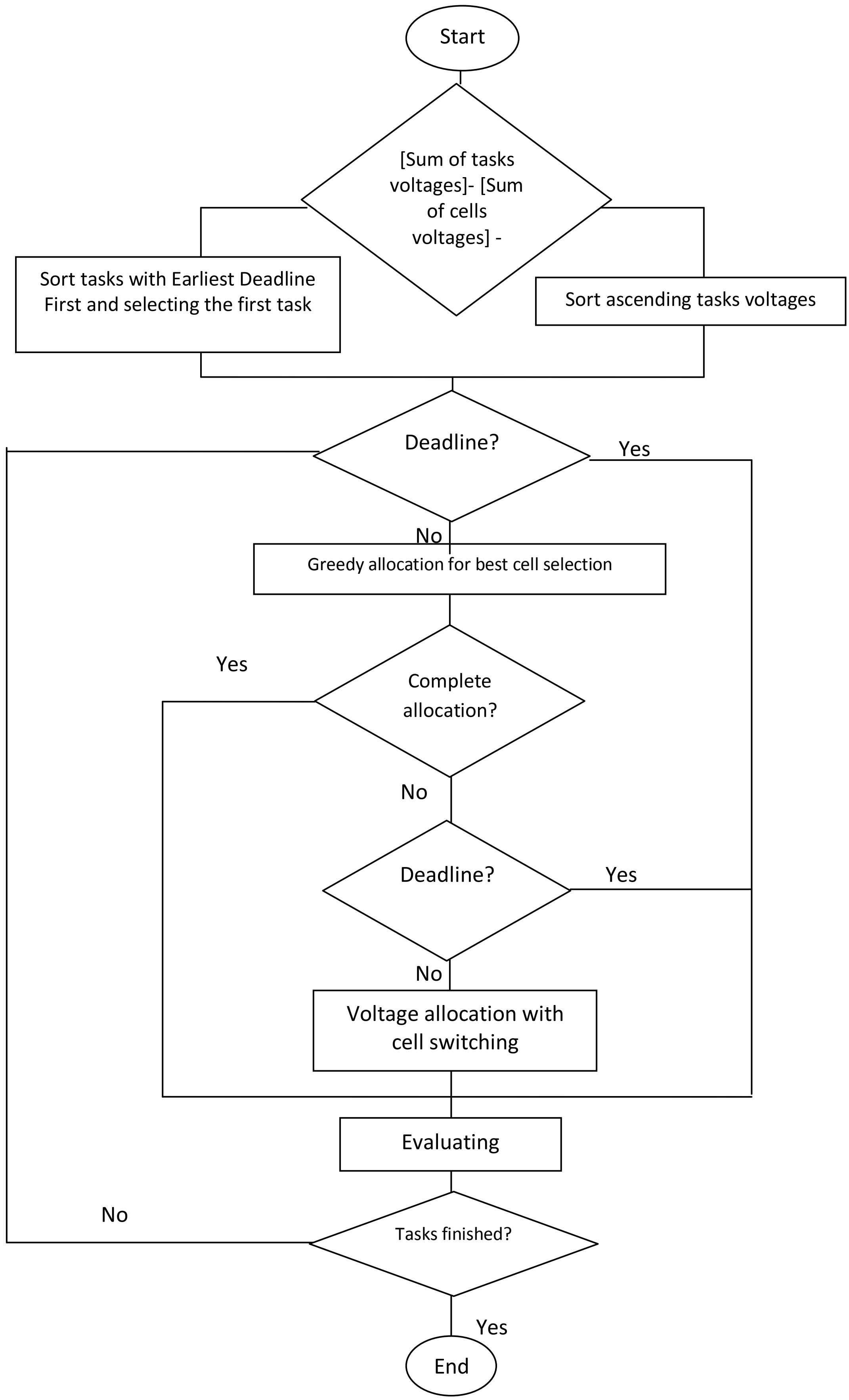


Figure.2GJTBS GJTBS flowchart

INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
www.ijrcar.com
Vol.3 Issue 12, Pg.: 5-15
December 2015

4. Simulation results

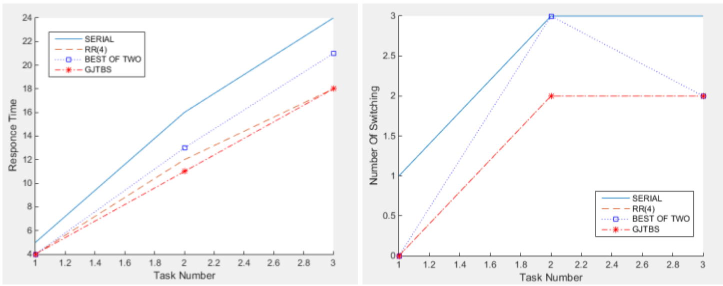
In this Section, we compare and Analyze the proposed algorithm GJTBS with three combination scheduling. Here we assumed the battery with 8 cells, like L Laptop battery. Cells number is variable. The proposed algorithm written in C++ with DEV software, then results presented in MATLAB 2014 Graphically. To measure suggested algorithm we consider three states: first, tasks voltage is equal to cells voltage, and a second tasks voltage is lower than cells voltage and the last one tasks voltage is more than cells voltage.

A. Tasks voltage is equal to cells voltage:

In this case, all tasks are executing, but the main point is How to consume battery with lower switching. Now, we are evaluating the algorithm with evaluation criteria.

To evaluate each of this criteria, we assumed constant 8 cells and three different scenarios with 3, 8, 12 tasks.

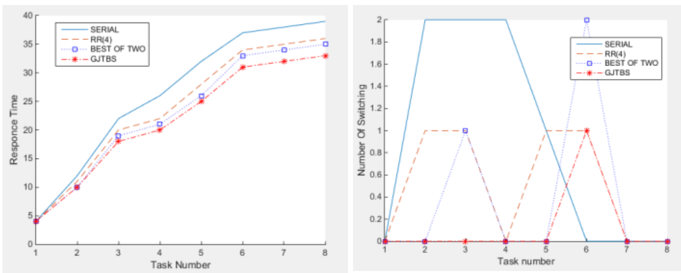
First scenario:



a b

Fig.3 tasks voltage is equal to cells voltage with 3 task and 8 cells
(a is Response time and b shows Number of switching)

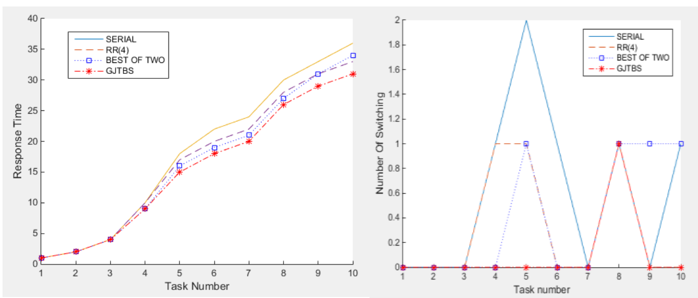
The second scenario:



a b

Fig.4 tasks voltage is equal to cells voltage with 8 task and 8 cells
(a is Response time and b shows Number of switching)

The third scenario:



a b

Fig.5 tasks voltage is equal to cells voltage with 10 task and 8 cells
(a is Response time and b shows Number of switching)

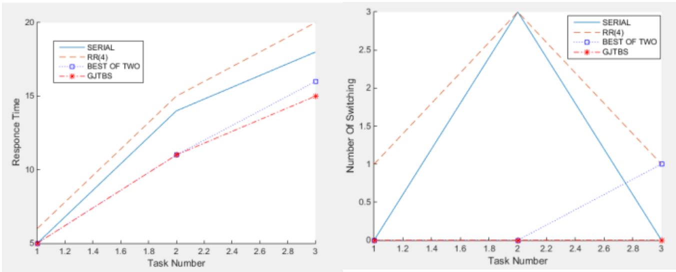
Vol.3 Issue 12, Pg.: 5-15
December 2015

B. Tasks voltage is lower than cells:

Second case happens when tasks voltage is lower than cells. Like the previous Case, as the battery voltage is more than tasks voltage, and the duration of the period is equal to the largest deadline, so task violation is impossible. Similar to the previous state, to evaluating GJTBS we use two criterion response time and the number of cell switching.

To evaluate the algorithms we used tasks randomly 3, 8 and 10 in 8 cells.

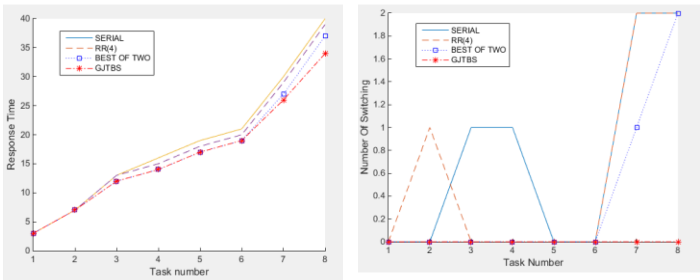
First scenario:



a b

Fig.6 tasks voltage is lower than cells with 3 tasks and 8 cells
(a is Response time and b shows Number of switching)

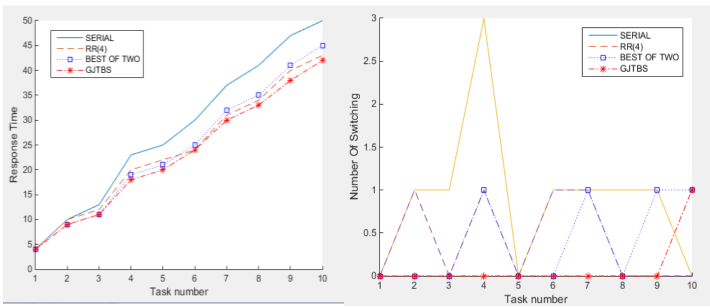
The second scenario:



A b

Fig.7 tasks voltage is lower than cells with 8 tasks and 8 cells
(a is Response time and b shows Number of switching)

The third scenario:



a b

Fig.8 tasks voltage is lower than cells with 10 tasks and 8 cells
(a is Response time and b shows Number of switching)

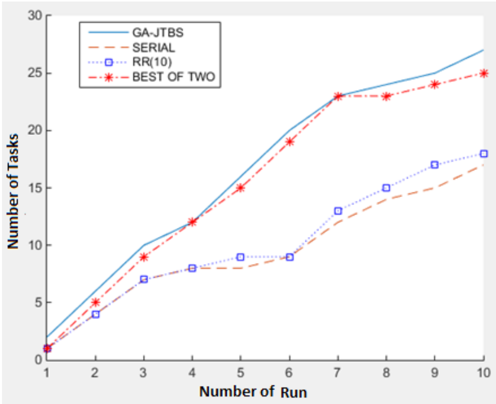


Fig.9 number of executed tasks in battery life time

C. Tasks voltage is more than cells voltage:

This happens when tasks voltage is more than cells voltage. In this case, we have undesirable task deadline violation. So here we have another important criterion named number of tasks which is done completely.

Note that, or tasks executed completely or executed not at all. And as we know execution of works after deadline is worthless, so there is no need to measure Time Response and the number of cell switching, because the number of completed task is most important factor. For this assessment, we simulated the algorithms based on 10 simulation periods of algorithms, and the number of different tasks from 3 to 30 times, with increasing the number of tasks in each period(shown in figure 9).

As shown in Figure 4-13, sequential and Round-Robin scheduling have a very close behaviour (sequential is developed algorithm of Round-Robin). The better behaviour of Round-Robin in comparison of sequential algorithm is for the recovery effect which happens on the rest time of battery. In other words, sequential algorithm doesn't use the rate capacity. The best selection algorithm, has been executed more works, and sometimes (in the 6th period) works exactly like the GJTBS (the proposed algorithm in this paper). But GJTBS's curve is above the all algorithms. This shows that the proposed algorithm can executed more tasks with considering their deadline.

5. Conclusion

In this paper we studied the joint task and battery scheduling in real-time systems. To this purpose first, we described the existing scheduling and at last we presented the greedy algorithm to select the Best battery Cell with the nearest deadline.

Goal of This Algorithm is to recover the battery lifetime of system, which leading to executing more tasks in the tasks lifetime. Here, we discussed a new algorithm to executing tasks in real-time systems with an intelligent battery.

Intelligent Batteries Contains Several Cells And the most important challenge in this type of battery is to providing appropriate algorithm to extend the lifetime of the system. Since the Battery Contains two Properties named Rate Capacity and charge Recovery, Scheduling task and battery Considers as np-hard problems. With attention to documentation Offered in this paper, we understand that parallel discharge of battery cells with Kinetic battery model leads to increasing lifetime.

Results of simulation, presents the better behaviour of proposed algorithm in comparison with three sequential, Round-Robin and best cell s election. Although the two Round-Robin and best cell selection algorithms are near optimized, but still they can prove it. The Results in this paper indicatives that suggested algorithm With Evaluation Criteria is near optimal approach and is far better than other algorithms.

Suggestions for future work

Since today battery is the most usual energy source, managing the battery usage is up to date problem. Using battery in battery security systems and even in wireless networks is very common. These systems need to small, light and programmable batteries to the future works. These types of batteries required algorithms which can schedule tasks in battery lifetime.

For the Next, it can use Genetic algorithms for Scheduling. And also in wireless networks, each sensor contains one small Battery. Scheduling is meaningless for one battery, but when we have a network of sensors, each sensors battery can consider as one cell and the proposed algorithm can use in it. Mostly, in these networks, there are routers which their duty is to transmitting collection of data, from Sensor nod To date Sink. To save more energy in all sensors, battery aware Routing for transmitting data must be done. Switching from one direction to other Direction Lead to charge Recovery and so it changes to battery scheduling problem.
Other thing that makes this more attractive is rechargeable batteries. In This Case not only charge recovery affects the charge of battery, but also the charge is increasing in the unit of time. In this case correct pattern for using battery makes the battery lifetime increase.
Using appropriate algorithms for battery scheduling leads batteries to become smaller, lighter and portable. Efficient Management of Enter loads To Systems, has the required Potential to Decrease battery Weight extremely. This is more Important and favourable between the soldiers that forced carrying Furniture, by their Organizations, Contains Batteries.

REFERENCES

[1] *Jongerden, R. And Haverkort, R. (2011). "Lifetime Improvement By Battery Scheduling", UKPEW White paper.*
[2] *Jongerde, M., Haverkor, B., Bohnenkam, B. and Katoe, J. (2009). "Maximizing System Lifetime By Battery Scheduling", EU-Project ICT-FP7-STREP.*
[3] *Jongerden, M., Mereacre, A., Bohnenkamp, H. (2010). "Computing Optimal Schedules Of Battery Usage In Embedded Systems", IEEE Transactions On Industrial Informatics, No. 3.*
[4] *Kopetz, H. (2011). "Real-Time Systems", Printed on acid-free paper of Springer, pp191-214.*
[5] *Jongerden, M. , Haverkort, B. , Bohnenkamp, H. , Katoen, J. P. (2009) "Maximizing system lifetime by battery scheduling" In: Proceedings of the 39th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2009). pp. 63–72. IEEE Computer Society Press.*
[6] *A. Hussein and I. Batarseh, (2011). "A Review of Charging Algorithms for Nickel and Lithium Battery Chargers" IEEE Trans. Veh. Technol., vol.60, no.3, pp.830-838.*
[7] *Jongerden, M. , Mereacre, A. , Bohnenkamp, H. , Haverkort, B. , Katoen,j. (2010) "Computing optimal schedules for battery usage in embedded systems". IEEE Transactions on Industrial Informatics 6(3), 276–286.*
[8] *Xiao, B. Shi Y.and He, L. (2010) "A universal state-of-charge algorithm for batteries" ACM/IEEE Design Automation Conference (DAC), pp.687-692, 13-18.*
[9] *Peng, O. Shouyi, Y. Leibo, L. Shaojun, W. (2010). "Battery Aware Tasks Allocating Algorithm for Multi-battery operated System", National S&T Major Project 2009ZX03006-004, NSFC Project (NO. 60803018) and Project (20070003098).*
[10] *Taesic, K. Wei, Q. and Liyan, Q. (2011). "Self-reconfigurable multicell batteries" Energy Conversion Congress and Exposition (ECCE), 2011 IEEE, pp. 3549-3555.*
[11] *Fox, M. and Long, D. Magazzeni, D. (2011), "Automatic Construction of Efficient Multiple Battery Usage Policies", Association for the Advancement of Artificial Intelligence.*
[12] *Yokoyama, T. Zeng, G. Tomiyama, H. and Takada, H. (2010), "Static task scheduling algorithms based on greedy heuristics for battery-powerd DVS Systems", IEICE Trans. on Information and Systems, vol. E93-D, no. 10, pp. 2737-2746.*
[13] *Qiu, M. Yang, L. T. Shao, Z. and Sha, E. (2010), "Dynamic and leakage energy minimization with soft real-time loop scheduling and voltage assignment", IEEE Transactions on Very Large Scale Integration Systems (TVLSI), vol. 18, no. 3, pp. 501–504.*
[14] *Wang T. and Cassandras, C. G. (2010), "Optimal discharge and recharge control of battery-powered energy-aware systems", in 49th IEEE Conf. on Decision and Control, Atlanta, GA, pp. 7513–7518.*
[15] *Gatzianas, M. Georgiadis, L. and Tassiulas, L. (2010), "Control of wireless networks with rechargeable batteries" IEEE Trans. on Wireless Communication, vol. 9, no. 2, pp. 581–593.*

Vol.3 Issue 12, Pg.: 5-15
December 2015

[16] *Chandy, K. M. Low, S. H. Topcu, U. and Xu, H. (2010), "A simple optimal power flow model with energy storage", in 49th IEEE Conf. on Decision and Control, Atlanta, GA, pp. 1051–1057.*

[17] *Liu, R. Sinha, P. and Koksal, C. (2010), "Joint energy management and resource allocation in rechargeable sensor networks", in IEEE INFOCOM 2010, San Diego, CA, pp. 1 – 9.*

[18] *Wang T. and Cassandras, C. G. (2010), "Optimal control of multi-battery energy-aware systems" Tech Report, Div. of Systems Engineering, Boston University, http://vita.bu.edu/renowang/Techreport2.pdf.*

[19] *Long, M. Magazzeni D. (2011). "Automatic Construction of Efficient Multiple Battery Usage Policies", In Proceedings of Int. Conf. on Automated Planning and Scheduling,(ICAPS), pp. 74–81.*

1st Mahta aali- *received her B.Sc. From IAU (Islamic Azad University Zanjan branch), Iran, a M.Sc. (2011) from department of computer, science and research branch, Islamic Azad university, Ardabil, Iran. All are in computer engineering. Her research interests are artificial intelligence, battery aware scheduling, energy aware scheduling and Computer systems.*

2nd Shahram Jamali- *is an assistant professor leading the Autonomic Networking Group at the Department of Engineering, University of Mohaghegh Ardabili. He teaches on computer networks, network security, computer architecture and computer systems performance evaluation. Dr. Jamali received his M.Sc. and Ph.D. degree from the Dept. of Computer Engineering, Iran University of Science and Technology in 2001 and 2007, respectively. Since 2008, he is with Department of Computer Engineering, University of Mohaghegh Ardabil and has published more than 50 conference and journal papers.*