Metrol. Meas. Syst., Vol. XX (2013), No. 1, pp. 87–98.

METROLOGY AND MEASUREMENT SYSTEMS
Index 330930, ISSN 0860-8229
www.metrology.pg.gda.pl

A NEW ELASTIC SCHEDULING TASK MODEL IN THE NODE OF A CONTROL AND MEASUREMENT SYSTEM

Wiesław Miczulski, Piotr Powroźnik

University of Zielona Gora, Faculty of Electrical Engineering, Computer Science and Telecommunications, Podgórna 50, 65-246 Zielona Góra, Poland (✉ W.Miczulski@ime.uz.zgora.pl, +48 68 328 2390, P.Powroznik@ime.uz.zgora.pl)

Abstract

The paper presents a new elastic scheduling task model which has been used in the uniprocessor node of a control measuring system. This model allows the selection of a new set of periods for the occurrence of tasks executed in the node of a system in the case when it is necessary to perform additional aperiodic tasks or there is a need to change the time parameters of existing tasks. Selection of periods is performed by heuristic algorithms. This paper presents the results of the experimental use of an elastic scheduling model with a GRASP heuristic algorithm.

Keywords: task scheduling, elastic scheduling task model, heuristic algorithms, measurement and control systems, time deadline fulfilment.

1. Introduction

The problem of scheduling exists in many areas of technology. The task scheduling theory is applied to appropriate areas of various tasks on specific devices (*e.g.*, processors, machines). This includes computer systems and networks and production systems and services, wherever there is a flow shop scheduling problem of information or materials [1–3]. The scheduling problem in this approach consists in assigning tasks to processors or machines, such that the criterion adopted obtains the highest efficiency in the use of devices with given constraints [4, 5]. These restrictions can be put in terms of availability or of required times for completion of tasks. While performing the tasks on processors or machines, retooling may also be required [6, 7]. For example, in a multiprocessor system the set-up process may consist of collecting the necessary input-output data from external systems, while in a production system it could be the matching of tools used to current production requirements [8, 9]. Such a task scheduling approach, as is here presented, is defined as the set of operations or activities executed with the intention of obtaining a given final product from the object (processor, machine or human) performing these operations [10]. Individual objects performing tasks can perform a number of other activities (tasks), which from the perspective of *e.g.* a manufacturing process are not relevant and not subject to scheduling analysis.

Another area for the application of scheduling theory is the analysis of time constraints through independent and most often periodic tasks carried out in the control and measurement system (CMS). These systems implement automatic measurement and control tasks in relation to a particular object or technological process.

In later parts of the work the application of the theory of task scheduling is presented in relation to tasks performed in a single uniprocessor node CMS. In the classical approach

Article history: received on Sep. 04, 2012; received in revised form on Dec. 17, 2012; accepted on Jan. 21, 2013; available online on Mar. 15, 2013; DOI: 10.2478/mms-2013-0008.

to such task scheduling a static model is used. In this model it is assumed that the node CMS tasks are periodic. Each task is described by the following time parameters [11]:

$$<C, T, D>, \tag{1}$$

where: C – the maximum time for the task, T – the period of occurrence of the task, D – relative time limit denoting the time limit within which the processor should finish the task.

The examination of the possibilities of completing all the tasks in the node CMS, with the given temporal parameters for each task, is performed by setting a condition for the use of the resources of node U CMS [11]:

$$U = \sum_{i=1}^{n} \frac{C_i}{T_i}, \tag{2}$$

where: i – index of task, n – number of tasks in a given CMS node.

The calculated value of $U \le 1$ designates the possibility to perform all the tasks on the data node CMS hardware resources. However, for $U > 1$ it is necessary to expand the node hardware resources to ensure implementation of all tasks. This solution has the disadvantage of increasing the cost of construction of the CMS node. In addition, the expanded resources of the node for most of the working time cannot be effectively utilized.

The application of CMS technology in electronics and information technology has made it possible to increase their functionality. At the same time it has resulted in the fact that in addition to periodic tasks with fixed time parameters the CMS node can perform tasks with changing time parameters. An example is the mobile robot, one of whose tasks is to avoid obstacles without suddenly stopping. In such a case it is necessary to increase the frequency of checking the distance to the obstacle which has been detected. There may also occur aperiodic tasks signalling *e.g*. the occurrence of a critical state in the operation of the object. Fig. 1 shows an example of a situation report for the aperiodic execution of a task in the node CMS.

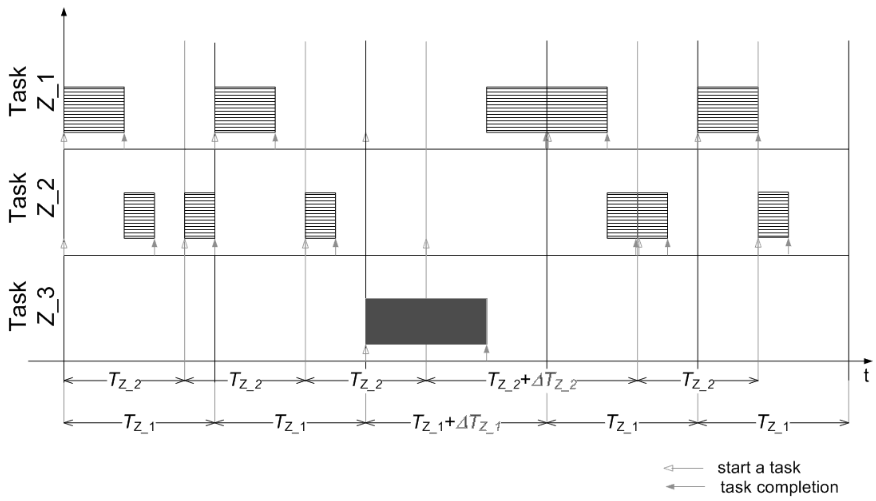

Fig. 1. Example of an elastic scheduling task model.

Tasks Z_1 and Z_2 are performed in a periodic manner for different settings T_{Z_1} and T_{Z_2}. It is assumed that task Z_1 has higher priority than the task of Z_2. For simplicity, Fig. 1 also assumes that for both tasks relative time constraints D are equal to the periods of occurrence of tasks T. The timing parameters of periodic tasks Z_1 and Z_2 are so selected that, in accordance with formula (2), the condition of resource use U is fulfilled. However, the application to the execution of aperiodic task Z_3, which has the highest priority, will lead to a breach of the condition of resource use U.

Metrol. Meas. Syst., Vol. XX (2013), No. 1, pp. 87–98.

The need to execute aperiodic tasks, or more frequently, to perform tasks such as one of the periodical ones, which have the highest priority, can lead to resource starvation for lower priority tasks. The resulting limitations of the static scheduling model (1), for no consideration or periodic tasks aperiodic inrush time-varying parameters, necessitated the need for an elastic scheduling model. Using this type of model will make it possible to fulfil the condition specified by (2), by modifying the period of occurrence T for all tasks. Fig. 1 illustrates this situation by increasing period T for tasks Z_1 and Z_2. Therefore, to ensure the implementation of tasks in a node at the CMS, or the application of an aperiodic task or tasks, or a periodic change of setting, the time parameters of a periodic task or tasks requires the designer to predict the CMS scenarios of such tasks. Once a scenario for the functioning of the SPS has been created for each case, an appropriate choice of a new set periods for the occurrence of T, in “online” or “offline” mode, will be made.

The few elastic scheduling task models in the published literature [12–14] have the following limitations:

– modification of the settings for all T periods of tasks executed in the CMS node is conducted in a proportionate manner, which can lead to a failure condition specified by (2);
– inability to decide which tasks should be subject to modified settings of periods T;
– inability of the designer of the CMS to identify the resource utilization U at a given level.

2. New elastic scheduling task model

Given the limitations of existing models of elastic scheduling, a new elastic scheduling model (ESTM) [15, 16] has been developed with the following parameters:

$$< C,\ T_{nom},\ D,\ T_{min},\ T_{max},\ wvt,\ U_{su} >, \tag{3}$$

where: T_{nom} – nominal period of the task, T_{min}, T_{max} – selection range of a new period for a task, wvt – weighting of the validity of the task, U_{su} – assumed value of the node resource use. Parameters C and D have the same meaning as in the model (1).

The development of the ESTM eliminates the above mentioned limitations of existing elastic scheduling task models. In ESTM, when selecting the new setting values for the periods of occurrence of tasks (T_{sel}), the assumed value of the CMS resource node use (U_{su}) and the weighting of the importance of the duties wvt are taken into account. The coefficient wvt points out which tasks should be subject to a modified adjustment period during task T at the expense of a smaller modification of setting periods of tasks T for the remaining tasks that are more important for the stability of the CMS. Tasks that are assigned with the highest coefficient wvt values have T_{sel} chosen in such a way that it is close or equal to the nominal values of the task T_{nom} specified in ESTM.

For each node of the task carried out in the CMS, the selection of new setting values T_{sel} periods (within limits T_{min} and T_{max}) belongs to the class of NP hard problems [16]. In this class of problems, for which finding the right solution (not necessarily optimal, but acceptable) is too expensive computationally, heuristic algorithms are used [17]. For further work on ESTM the following algorithms have been chosen: evolutionary, simulated annealing, tabu search, ants, A*, GRASP. It is assumed that the values selected by the heuristic algorithm T_{sel} do not necessarily constitute an optimal solution, but must take into account the assumed weighting importance of the tasks wvt and U_{su}. The algorithms, evolutionary, simulated annealing and tabu search, were chosen, among other reasons, because of their prior use in the scheduling of tasks, albeit in terms of allocation of tasks to machines [4]. The other three algorithms (ants, A* and GRASP) were selected because of their mode of action during the search for solutions to the difficult problem of NP, which consists in modeling solutions in the search area in the form a graph. The problem of selecting in the ESTM the T_{sel} periods can also be represented by a graph, as is, for example, shown for the three tasks in Fig. 2. A set of periods T_{sel} is followed consecutively for each of the tasks in the range from T_{min} to T_{max}.

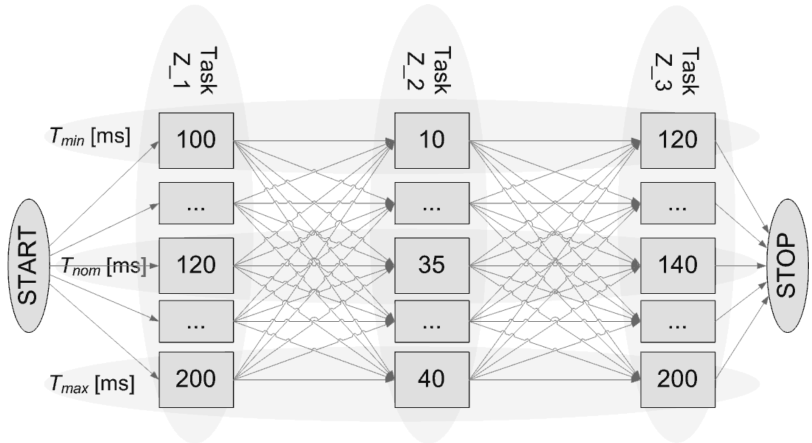

Fig. 2. Sample graph for selecting settings T_{sel} in ESTM for three tasks.

In order to ensure the proper operation of heuristic algorithms in ESTM, a model for the assessment of the solution (Or) of selecting new settings T_{sel} periods for all tasks [18] has been developed:

$$Or=\begin{cases}\displaystyle\sum_{i=1}^{n}\left(\frac{\left(T_{nom_i}-T_{sel_i}\right)^2}{\sum_{j=1}^{n}\left(T_{nom_j}-T_{sel_j}\right)^2}\cdot\frac{wvt_i}{\sum_{k=1}^{n}wvt_k}\right) & \text{, when } \displaystyle\sum_{j=1}^{n}\left(T_{nom_j}-T_{sel_j}\right)^2>0\\ 0 & \text{, when } \displaystyle\sum_{j=1}^{n}\left(T_{nom_j}-T_{sel_j}\right)^2=0\end{cases}\qquad(4)$$

where: j, k – index of task.

The developed model (4) was used as the objective function in heuristic algorithms, whose value was minimized in respect to the selection of T_{sel}. This is reflected in its recording of *wvt* coefficients for all tasks. Each of the heuristic algorithms was supplemented by checking the condition of use of resources U for newly designated T_{sel} periods. The calculated U-value is compared with the assumed value of ESTM U_{su}. In the case where the value of U is less than or equal to U_{su}, the heuristic algorithm is terminated. Otherwise, a new heuristic algorithm T_{sel} periods is set for all tasks. In addition, for part of the heuristic algorithms, in order to ensure proper operation in ESTM, it was necessary to carry out simulation studies to enable the selection of appropriate methods and parameter values for these algorithms.

For all the above mentioned heuristic algorithm simulation studies, the aim was to compare the properties of these algorithms in respect to an evaluation of their applicability for the determination of the settings in ESTM T_{sel} periods. Simulation studies were carried out in a Matlab environment for different numbers of tasks in the CMS node. Each test was repeated 1000 times, with the two designated parameters (Lor_{avg} and u) which formed the basis of comparison of selected properties of heuristic algorithms. The parameter Lor_{avg} is calculated as the average of the number of values of solutions (Lor) given in subsequent repetitions of these algorithms. The parameter Lor_{avg}, which was chosen because it is the calculated value of the function Or, enabling the assessment of the T_{sel} tuning solutions, is the most costly operation during operation of the heuristic algorithm. Checking how well the value of T_{sel} fits

Metrol. Meas. Syst., Vol. XX (2013), No. 1, pp. 87–98.

the values assumed in the ESTM values obtained by calculating the U_{su} of another parameter u, which is the relative resource use of the CMS node, is defined by the formula:

$$u = \left|\frac{U_{su} - U_{avg}}{U_{avg}}\right| \cdot 100\%, \tag{5}$$

where: U_{avg} – U mean value of all iterations of the algorithm simulation tests.

The results of simulation studies, with descriptions of several algorithms for matching the ESTM, are published in [19–22]. However, reference [16] presents a comparison of selected properties of heuristic algorithms made on the basis of the adopted parameters u and Lor_{avg}. Simulation studies have shown that all the analyzed heuristic algorithms can be used in the ESTM. The most preferred algorithm for use in the ESTM is the GRASP algorithm, because the selected settings of T_{sel} are obtained at the lowest cost of calculation and it also gives the best reflection of the rate of resource use U_{su}. A block diagram of the GRASP algorithm used in the ESTM is shown in Fig. 3.

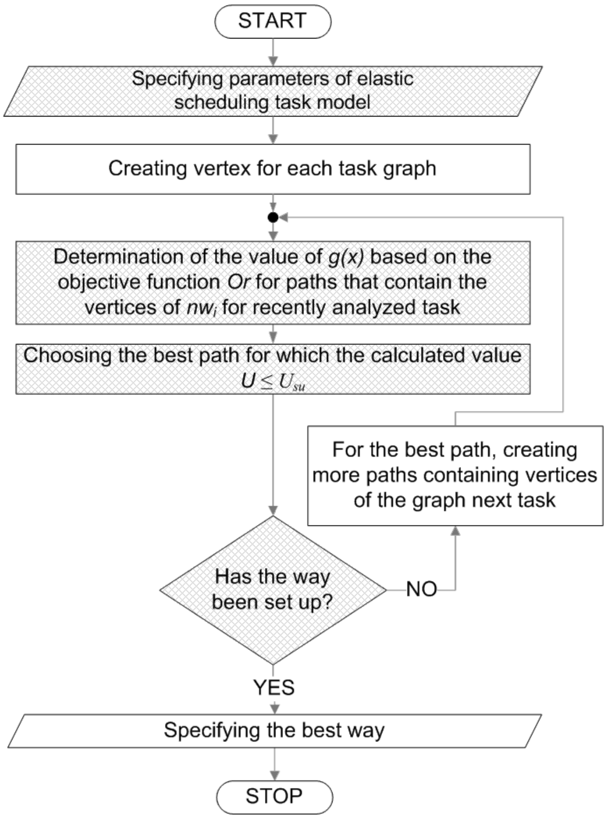

Fig. 3. Block diagram of the GRASP algorithm ESTM.

The GRASP algorithm in the ESTM begins by creating graph peaks nw_i according to the method of creating a graph shown in Fig. 2. T_{sel} selection for each task is based on the values of $g(x)$, which are the already chosen values of the setting periods T_{sel} and T_{max} values, for those tasks which have not yet had selected the new setting periods. Calculation of numerical values of $g(x)$ is made on the basis of the objective function Or (4). In the event that the value calculated from equation 2 of $U > U_{su}$ T_{sel} during the selection of setting T_{sel} for i, then the task is discarded. The GRASP algorithm terminates only when periods T_{sel} have been selected for all tasks (the path will be created).

4. Experimental verification of the GRASP algorithm ESTM

The aim of this experimental study was to examine whether the ESTM with the GRASP algorithm can give, in an "online" or "offline" mode, assurance of implementation of tasks in the CMS node in the case of notification of aperiodic tasks or tasks with periodically changing time parameters.

The study was conducted in laboratory conditions with the following assumptions: The CMS includes two nodes, node 2 and node 3 (Fig. 4); in node 2 a program implementing the ESTM with the GRASP algorithm is installed. This program was developed on the platform of JavaTM 2 Micro Edition (J2METM) in the NetBeans IDE environment with Java ME SDK 3.0. The J2METM platform was chosen because of the dedicated nature of its application in devices with limited memory and processing power. Node 3 carries out the tasks with the assumed time parameters. In the case of notification for carrying out aperiodic tasks in node 3 or the need to change the parameter setting the time for the periodic tasks of the node, it sends a query to node 2 to enter a new setting T_{sel} for all tasks performed in the third node.

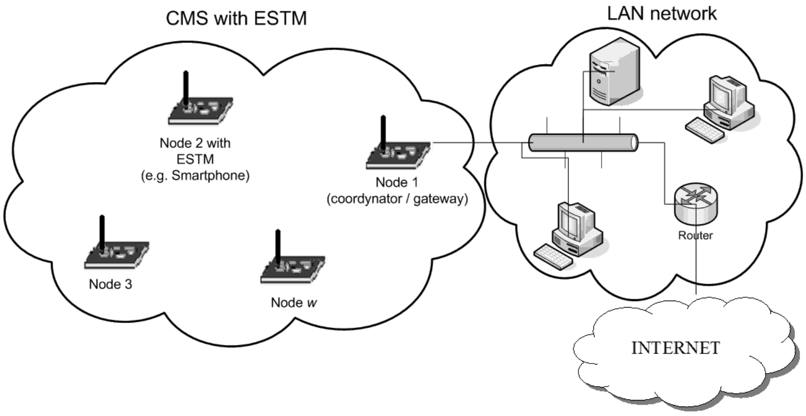

Fig. 4. Sample of CMS with ESTM.

Experimental studies were carried out on four node 2 configurations (*nc*2) equipped with a specific type of hardware and communication with node 3 (Table 1). *nc*2_1 configuration. There was chosen for the *nc*2_1 configuration a hardware configuration corresponding to the latest production of wireless sensor nodes, IEEE 802.15.4/ZigBee with powerful hardware resources. Configurations *nc*2_3 and *nc*2_2 were chosen such that the PC, implementing the wired and wireless communications, was more efficient in terms of the hardware configuration than that used in the *nc*2_1 (smartphone).

Table 1. Configuration summary of node 2.

nc2	Type of communication	Equipment of a node
1	wireless	Smartphone Nokia C7 – 00 processor ARM1176 (TSMC 65GP) 680 MHz Symbian^3
2	wireless	computer Intel Core i3-540 4GB PC3-10600 (DDR3-1333) Windows7 Professional z Service Pack1 64-bit WD5000AAKS 7200 rpm
3	wire	
4	wire	computer Intel Celeron 850 MHz 512 MB DDR-266 (PC-2100) WindowsXP Professional with Service Pack3 32-bit ST340810A 5400 rpm

The performance of the computer in the configuration of *nc*2_4 was comparable to the performance of the hardware in the *nc*2_1 configuration, while node 3 was a PC. This computer was connected by cable (standard 100BASE-TX) to a Linksys router WRT54GC-EU, which would send a request designating new settings T_{sel} to node 2, by wired or wireless

Metrol. Meas. Syst., Vol. XX (2013), No. 1, pp. 87–98.

means. Wired or wireless communication between node 2 and node 3 was conducted in accordance with the UDP protocol. The exchange of data between CMS nodes used a data encoding format developed in the form of a frame consisting of a specific number of bytes (Table 2 and 3). For each task, along with the ESTM parameters, there is also sent the number of settings for the choice of T_{sel} period (lz). In the last data field (Table 3), on two bits, information was placed on how to obtain the setting T_{sel} (appointed by the GRASP algorithm, or read from memory) and the writing to memory of the settings T_{sel}.

Table 2. Meanings of each field in the frame transmitted from node 3 to node 2.

The number of fields	**The number of bytes**	**Meaning bytes**
1	2	Number of tasks (n)
2	4	U_{su}
3	28	$C, D, T_{nom}, T_{min}, T_{max}, wvt, lz$ – ESTM parameters for the task 1
...	...	...
$n + 2$	28	$C, D, T_{nom}, T_{min}, T_{max}, wvt, lz$ – ESTM parameters for the task n

Table 3. Meanings of each field in the frame transmitted from node 2 to node 3.

The number of data fields	**The number of bytes**	**Meaning bytes**
1	2	Number of settings T_{sel}
2	4	T_{sel} – for the task 1
...	...	...
$n + 1$	4	T_{sel} – for the task n
$n + 2$	1	Bit 0 – if setting a record obtained T_{sel}. Bit 1 – if the reading obtained T_{sel} settings. If you do not read the following appointment setting T_{sel} GRASP algorithm.

As described in Table 2, a maximum frame length equal to 7006 bytes for 250 tasks (accepted maximum number of tasks in node 3 CMS) was created. However, during the return of designated setting T_{sel} by the GRASP algorithm a frame was created with the maximum length equal to 1003 bytes for 250 tasks. In both cases, the number of transmitted data does not require a high-bandwidth communication channel to be used.

Experimental studies consisted of measuring the time tk, which was calculated from the moment of transmission by node 3 to node 2 the T_{sel} request for new settings, until the designated setting T_{sel} was received back by the GRASP algorithm. Time tk was read at node 3 with a counter counting clock pulses. For each number of tasks n (5, 15, 25, 50, 75, 100, 150 and 250) implemented in node 3 the average communication time was calculated for (tk_{avg}) resulting from 1000 iterations of the tests performed. Time values were determined for four tk_{avg} configurations of node 2 (Table 1), in which the following activities were carried out:

1. Determination of setting T_{sel} without trying to read them first and without saving them ($r0c1w0$).
2. Determination and saving settings T_{sel} ($r0c1w1$).
3. Reading T_{sel} settings and saving them again ($r1c0w1$).
4. Reading settings T_{sel} ($r1c0w0$).

Figure 5 shows the results of the calculated values tk_{avg} for a certain number of tasks n, which are obtained when calculating the adjustment T_{sel} through the ESTM with a GRASP algorithm at node 2 $nc2\_1$ CMS configuration. The research shows that:

1. tk_{avg} time values increase with an increasing number of tasks performed on a CMS node n, when it is required to designate a setting T_{sel} ($r0c1w0$, $r0c1w1$).

2. For a small number of tasks ($n \leq 25$) it is preferable to designate T_{sel} settings without saving. Reading settings T_{sel} ($r1c0w0$) is longer than their re-designation ($r0c1w0$).
3. From a comparison of characteristics for read-only setting T_{sel} ($r1c0w0$) with the characteristics of the read and write of ($r1c0w1$) T_{sel} it turns out that the saving of T_{sel} results is a long operation. This is due to the properties of disk storage used in the smartphone Nokia C7–00.

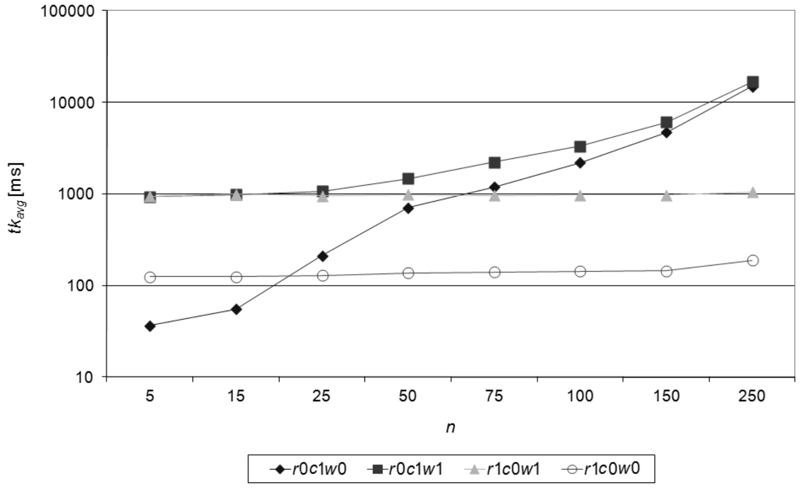

Fig. 5. tk_{avg} time values for different numbers of tasks n, for the configuration $nc2\_1$.

The calculated time values of tk_{avg} for a specific number of tasks n performed on a node 2 CMS with a $nc2\_2$ configuration, shown in Fig. 6, indicate that:

1. The use of a more efficient configuration of $nc2\_2$ (PC with a faster disk memory) relative to the configuration of $nc2\_1$ (Smartphone) reduced the effect of read and write operation T_{sel} settings ($r1c0w1$) for tk_{avg} time.
2. Together with the increase in the number of tasks n, for which was designated setting T_{sel}, it was more useful to save and then read the settings T_{sel} ($r1c0w1$), or just read them ($r1c0w0$), than to determine them.

Calculated values of time tk_{avg} for a specific number of tasks n performed on a node 2 CMS with a $nc2\_3$ configuration reached very similar results to those for the configuration $nc2\_2$. This means that for the same hardware configuration the means of communication between CMS nodes (wireless in a $nc2\_2$ or wired in a $nc2\_3$) had no significance on the performance of the time value tk_{avg}.

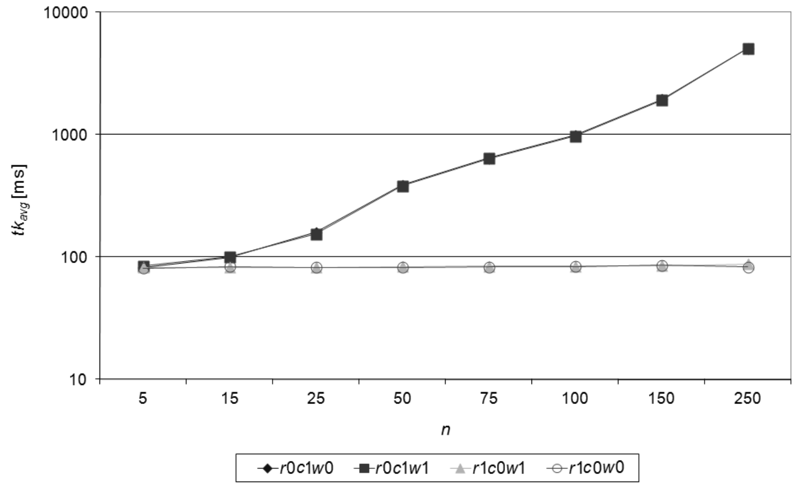

Fig. 6. tk_{avg} time values for different numbers of tasks n, for the configuration $nc2\_2$.

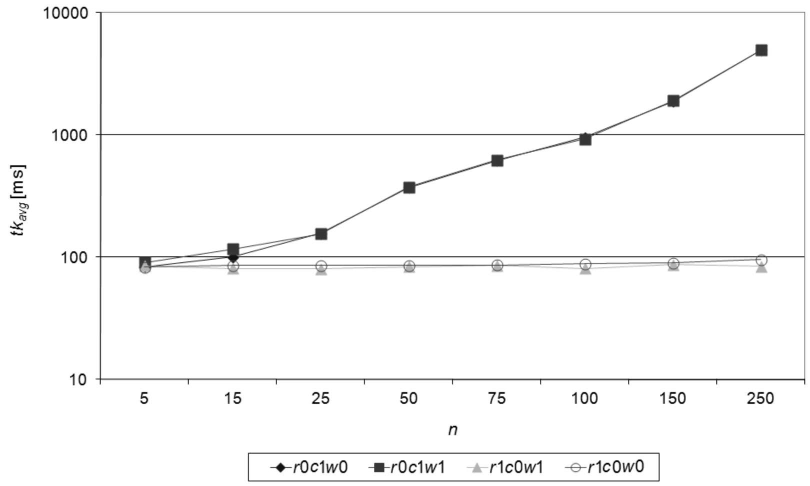

Fig. 7. tk_{avg} time values for different numbers of tasks *n*, for the configuration *nc*2_4.

Figure 7 shows the results of the calculated values tk_{avg} for a specific number of tasks *n*, which were obtained when calculating the adjustment T_{sel} by the ESTM with the GRASP algorithm in a node 2 CMS with a *nc*2_4 configuration. The research shows that:

1. The use of different data storage, despite similar power of computational units in *nc*2_4 *nc*2_1 configurations, has an impact on the value of time tk_{avg} during the operation of writing to memory: (*r*0*c*1*w*1) and (*r*1*c*0*w*1).
2. The use of less efficient equipment in configuration *nc*2_4 than in *nc*2_3 extends the time setting T_{sel} in the configuration setting *nc*2_4 in comparison with *nc*2_3.

The influence of the configuration of the CMS node 2 on tk_{avg} time calculation results for various numbers of tasks *n* in determining the setting T_{sel} (*r*0*c*1*w*0) by the ESTM with the GRASP algorithm is shown in Fig. 8. The presented results indicate that:

1. The configuration *nc*2_1 had the fastest T_{sel} determined settings for twenty-five tasks relative to the configurations *nc*2_2 and *nc*2_3, which used equipment with greater efficiency than the *nc*2_1 configuration.
2. For two configurations *nc*2_1 and *nc*2_4, similar in terms of hardware, T_{sel} setting values for each analyzed number of tasks *n* were determined more quickly in the case *nc*2_1 configuration.

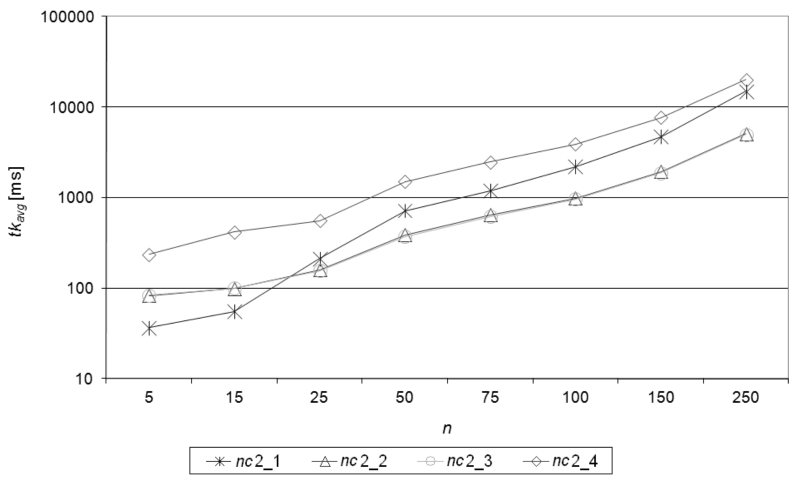

Fig. 8. tk_{avg} parameter values for different numbers of tasks *n*, when T_{sel} settings are determined without saving them and without attempts at prior reading (*r*0*c*1*w*0).

In both cases, the achievement of lower values of tk_{avg} for the configuration *nc*2_1 is the result of the operating system and Symbian^3 in the smartphone Nokia C7–00. Symbian^3 is a dedicated system for the device in which, with due regard to energy consumption, only a few short and less complex processes are running in contradistinction to Windows. Not without significance, there may also be an influence from the implementation of the Java virtual machine, on which midlet is run , designating the setting T_{sel} in the ESTM with the GRASP algorithm.

5. Conclusions

In the first part of the paper a new elastic scheduling task model has been presented which allows, in the case of the occurrence in the CMS node of aperiodic tasks or periodic tasks with changing time parameters, the selecting of new setting periods for the occurrence of tasks (T_{sel}) for all tasks in the CMS node. In the ESTM developed in this paper the use of heuristic algorithms is proposed for the determination of T_{sel}. On the basis of simulation studies, allowing a comparison of selected properties of heuristic algorithms, the use of the GRASP algorithm in the ESTM has been proposed.

In the second part of the article the preliminary results of experimental application of the GRASP algorithm in the ESTM node CMS has been presented. This stage of the study was conducted in the laboratory and limited to the designation of period T_{sel} settings for only one node. Equally, no account was taken of the impact of additional disturbances occurring in wireless networks. Such an experimental verification has confirmed the possibility of applying the ESTM with the GRASP algorithm for the determination of the settings T_{sel} in one of the CMS nodes.

The presented results of the experimental studies have shown a particularly high dependence of the time tk_{avg} on the hardware and software configuration of the CMS node. In case of the hardware configuration of the CMS node the performance of the memory used may be important when reading and writing the ESTM previously designated by the GRASP algorithm T_{sel} settings. Not without significance is the fact that the software configuration of the CMS system node implementing support for all the tasks executed in the node, among others, the maintenance tasks for the program implementing the ESTM.

On the basis of the analyzed configurations it can also be noted that a small number of tasks (up to 25) designating T_{sel} settings can be done in the "online" mode on hardware and software resources not worse than those in the *nc*2_1 configuration. Getting responses from a designating node in the "online" mode for a new set of periods T_{sel} by the ESTM with the GRASP algorithm, in not more than 0.2 seconds, allows the use of such a solution also in wireless sensor networks, IEEE 802.15.4/ZigBee. For a greater number of tasks the use of more efficient hardware configurations to determine the setting T_{sel} will not always be necessary, since a node implementing the CMS ESTM with the GRASP algorithm can be made to read from memory earlier settings of T_{sel} ("offline" mode). The obtained results indicate that the "offline" mode allows retrieval of a new set of periods T_{sel} in not more than 0.2 s.

Acknowledgements

This scientific work was funded under the grant for young researchers and doctoral students in the years 2011–2012 (number 506-06-01-01).

Metrol. Meas. Syst., Vol. XX (2013), No. 1, pp. 87–98.

References

[1] Janiak, A., Kołodka, P., Krysiak, T. (2010). Scheduling jobs with changeable job values and different release dates – solution algorithms. *Electrical Review*, 9, 91–96.

[2] Allahverdi, A., Cheng, C.T., Kovalyov, M.Y. (2008). A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, 187, 985–1032.

[3] Raut, S., Swami, S., Gupta, J. N. (2008). Scheduling a capacitated single machine with time deteriorating job values. *Int. J. Production Economics*, 114, 769–780.

[4] Janiak, A. (1999). *Selected problems and scheduling algorithms and resource allocation*. Akademicka Oficyna Wydawnicza PLJ.

[5] Bellanger, A., Oulamara, A., Kovalyov M.Y. (2010). Minimizing total completion time on a batching machine with job processing time compatibilities. *Electronic Notes in Discrete Math.*, 36, 1295–1302.

[6] Janiak, A., Krysiak, T. (2007). Single processor scheduling with job values depending on their completion times. *Springer*, 129–138.

[7] Tang, L., Zhao, Y. (2008). Scheduling a single semi-continuous batching machine. *Discrete Applied Mathematics, Omega*, 36, 992–1004.

[8] Janiak, A., Janiak, W.A., Januszkiewicz, R. (2009). Algorithms For Parallel Processor Scheduling With Distinct Due Windows And Unit-Time Jobs. *Bulletin Of The Polish Academy Of Sciences Technical Sciences*, 57.

[9] Błażewicz, J., Kovalyov, M. Y., Machowiak, M., Trystram, D., Węglarz, J. (2006). Preemptable Malleable Task Scheduling Problem. *IEEE Transactions On Computers*, 55, 486–490.

[10] Winczaszek, M. (2006). *Some scheduling problems with due windows assignment. Ph.D. Thesis*. Politechnika Wrocławska.

[11] Liu, C., Layland, J. (1973). Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. *Journal of the Association for Computing Machinery*, 1, 46–61.

[12] Cervin, A., Eker, J. (2000). Feedback Scheduling of Control Tasks. *39th IEEE Conference on Decision and Control*, Sydney, Australia.

[13] Buttazzo, G., Lipari, G., Caccamo, M., Abeni, L. (2002). Elastic scheduling for flexible workload management. *IEEE Transactions on Computers*, 51(3), 289–302.

[14] Chantem, T., Hu, S., Lemmon, M. (2006). Generalized Elastic Scheduling. *Real-Time Systems Symposium, 27th IEEE International Volume*, 236–245.

[15] Powroźnik, P. (2009). Elastic Scheduling Task Model with Evolution Algorithm in the Control Measurement Systems. *Electrical Review*, 2, 79–82.

[16] Miczulski, W., Powroźnik, P. (2011). Analysis of the properties of some heuristic algorithms used in Elastic Task Model Scheduling. *Electrical Review*, 9a, 107–111.

[17] Mikołajczak, B., Stokłosa, J. (1986). *Computational complexity of algorithms*. Wydawnictwo Politechniki Poznańskiej.

[18] Powroźnik, P. (2012). Influence of the parameters of fitness function for the periods set selection in elastic task model scheduling. *Studia Informatica*, Gliwice, 33(3A), 101–110.

[19] Powroźnik, P. (2009). Selection analysis of evolution algorithm of chosen initially parameters in task scheduling elastic model. *In Proc. XVII Polish Conf. Modelowanie i symulacja systemów pomiarowych, XVII Sympozjum*. Krynica, Polska, 127–134.

[20] Powroźnik, P. (2010). Verification of correct operation of tabu search algorithm in elastic task model sched-uling in control and measurement systems. *In Proc. VIII Polish Conf. Systemy pomiarowe w badaniach naukowych i w przemyśle SP'2010 Łagów, Konferencja Naukowa SP 2010*, Łagów, Polska, 21–23 czerwca 2010, 131–134.

[21] Powroźnik, P. (2010). Elastic Task Model Scheduling based on Simulated Annealing in node of Control Measurement Systems. *In Proc. V Polish Conf. Kongres Metrologii – KM 2010, V Kongres Metrologii, Łódź, Polska*, 100–101.

[22] Powroźnik, P. (2011). Properties of some heuristic algorithms: A*, GRASP and ants algorithm used in elastic task model scheduling. *In Proc. XVIII Polish Conf. Modelowanie i symulacja systemów pomiarowych, XVIII Sympozjum. Krynica, Polska,* 123–130.