

УДК 004.414.22/004.051/519.687.5

С. С. АЛЬ-КХШАБ

*Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ», Украина*

## ПРИНЦИП АДАПТИВНОГО ПЛАНИРОВАНИЯ ЗАДАЧ В УСЛОВИЯХ АВТОНОМНОЙ РАБОТЫ ПОРТАТИВНОГО КОМПЬЮТЕРА

*Рассмотрены принципы управления производительностью современных процессоров, в основе которых энергосберегающая технология динамического масштабирования напряжения и частоты процессора. Показано, что для эффективного применения динамического масштабирования необходимо синтез расписания задач с учетом ограничений мягкого реального времени. Синтез расписания основан на решении двухкритериальной оптимизационной задачи с выделением множества Парето-оптимальных решений. К числу критериев относятся минимумы штрафов за снижение качества обслуживания и потерю продолжительности автономной работы с текущим профилем электропотребления. Сужение множества Парето-оптимальных решений возможно на основе учета предпочтений пользователя.*

**Ключевые слова:** адаптивность, динамическое изменение напряжения и частоты процессора, мягкое реальное время, Парето-оптимальное решение, процессор, оптимизация, энергосбережение.

### Введение

Большинство из современных портативных компьютеров (ноутбуков и нетбуков) по вычислительной мощности практически не уступают стационарным компьютерам и одновременно с этим обладают такими свойствами, как портативность и переносимость. Но эти плюсы влекут за собой ряд довольно значительных проблем: отвод излишков тепла, образуемого при работе таких компонентов как процессор, видеокарта, малый срок работы устройства без подключения к сети электропитания. Согласно отчету о потреблении энергии бытовой электроникой в США в 2013 году, представленном Fraunhofer USA Center for Sustainable Energy Systems [1], в среднем такое портативное устройство потребляет за год 53 кВт\*час, при этом количество портативных компьютеров в США уже в 2011 году превысило количество стационарных, а в дальнейшем разрыв продолжает увеличиваться.

Что касается проблемы быстрого разряда аккумулятора при работе без подключения к сети электропитания, то помимо неудобств в использовании, быстрый разряд несет за собой еще одну проблему – необходимость утилизации элементов питания после окончания сроков их эксплуатации. Уменьшение суммарного энергопотребления мобильных устройств способствует не только увеличению времени разряда батареи мобильного устройства, но и продляет срок жизни аккумулятора, что в свою очередь снижает общие затраты на утилизацию элементов питания. Проблема имеет два взаимодополняющих

пути решения – это применение новых технологий для создания аппаратных компонент (процессора, видеокарты, аккумулятора и др.) и внедрение новых подходов к разработке энергосберегающего программного обеспечения (ПО). В контексте таких подходов можно отметить следующее.

Работа процессора с максимальной производительностью влечет быстрый разряд аккумулятора. В то же время, как показывает практика, нет постоянной необходимости в таком использовании современных процессоров. Большинство приложений не требуют мгновенного выполнения, а должны быть завершены за определенный промежуток времени. Так, к примеру, воспроизведение видеоизображения представляет собой последовательную обработку пакетов, поступающих через строго определенные промежутки времени. Соответственно необходимо успеть обработать поступивший пакет до получения следующего и нет смысла завершать задачу заранее. При оптимизации энергопотребления критерием эффективности можно считать затраченную энергию на выполнение всех задач в строго определенных временных интервалах. Процессор не должен активно работать при отсутствии задач.

### Анализ публикаций

На аппаратном уровне разработчики микропроцессорной техники предоставляют возможность управления производительностью. Например, SpeedStep – энергосберегающая технология Intel, в основе которой лежит динамическое изменение час-

тоты и напряжения питания процессора [2]. Известны аналогичные энергосберегающие технологии от AMD. PowerNow! - технология, разработанная для применения в версиях процессоров K6-2+, K6-III+ и Athlon, которые устанавливаются в ноутбуки. Тактовая частота и напряжение питания процессора автоматически снижаются, когда компьютер простаивает или недостаточно загружен. Это позволяет снизить энергопотребление (увеличить время работы от батарей) и уменьшить тепловыделение.

В основе этих энергосберегающих технологий - метод динамического изменения напряжения (Dynamic Voltage Scaling, DVS) [3]. Подаваемое напряжение определяет частоту работы процессора (задержки между выполнением инструкций), а значит, воздействует на энергопотребление. При этом переключения напряжения являются энергоемкими и достаточно продолжительными (дорогими) переходными процессами. Соответственно, выявление оптимальной частоты работы процессора для выполнения поступающих задач является актуальной проблемой. Алгоритмы DVS представляют собой планировщики напряжения в реальном времени, которые используют известную зависимость потребляемой процессором мощности от режимных параметров [4]:

$$P = ACV^2f + \tau AVI_{\text{short}}f + VI_{\text{leak}}, \quad (1)$$

где первое слагаемое измеряет динамическое энергопотребление, вызванное зарядом и разрядом емкостной нагрузки на выходе каждого ключа, которое равно произведению ёмкости – С, квадрата напряжения питания - V, частоты процессора f, и коэффициента А, характеризующего активность ключей в системе;

второе – это мощность, затрачиваемая в результате тока короткого замыкания, который проходит в момент переключения логического элемента;

третье – это потери из-за тока утечки.

В современных схемах, первое слагаемое доминирует, поэтому уменьшение напряжения питания является наиболее эффективным способом снижения потребления энергии. Квадратичная зависимость энергопотребления от напряжения означает, что экономия может быть значительной. К сожалению, эта экономия происходит за счет производительности, поскольку максимальная рабочая частота равна:

$$f_{\text{max}} = \frac{k(V - V_{\text{threshold}})^2}{V}, \quad (2)$$

где k - коэффициент,

$V_{\text{threshold}}$  - пороговое напряжение ключа.

Таким образом, скорость работы примерно пропорциональна корню кубическому от мощности [5, 6]:

$$f = k_f \sqrt[3]{P}. \quad (3)$$

Максимальная производительность обеспечивается при наибольшей частоте, но при этом процессор потребляет максимальную мощность. Уменьшив частоту и, особенно, напряжение питания, можно снизить потребляемую мощность и обеспечить более длительную работу ноутбука (нетбука) от батареи.

В литературе можно найти различные вариации модели 1-3, используемой для планирования задач [7, 8]. Обычно предполагается модель неограниченной скорости работы процессора, когда процессор может работать с любой скоростью из диапазона [0,1). В некоторых публикациях предполагается ограниченная модель скорости, когда допустимые скорости лежат в некотором интервале, либо принадлежат конечному множеству дискретных значений [9].

Распространено два основных принципа планирования [10].

1. На основе оценки загруженности процессора на определенном временном интервале принимается решение об изменении подаваемого напряжения, при этом на изменение интенсивности поступающих задач система может реагировать с запаздыванием. Адаптивные алгоритмы такого рода используют критерий среднего риска или близкие к нему.

2. При поступлении в систему очередной задачи выставляется определенное значение напряжения, гарантирующее выполнение всех задач, что может приводить к нежелательным скачкам напряжения.

По критерию учитываемой информации можно выделить 2 типа алгоритмов.

1. Online-алгоритмы принимают решения только на основе доступной во время выполнения информации, то есть не зависят от информации о будущем поведении системы, не располагают какими-либо априорными знаниями о входных характеристиках потока задач.

2. Offline-алгоритмы располагают полной информацией о потоке задач, поэтому представляют скорее только теоретический интерес.

В общем случае применяют методы, основанные на минимизации той или иной функции стоимости [11], когда каждая задача определяет некоторую стоимость, которая учитывается в случае невыполнения задачи до заданного времени. Соответствен-

но, оптимизация сводится к минимизации функции потерь. По критерию оптимизации известные алгоритмы энергосберегающего планирования разделим на 3 группы.

1. Минимизация среднего времени отклика. Pruhs, Uthaisombut и Woeginger предложили эффективный Off-line алгоритм планирования, который минимизирует среднее время выполнения с ограничением на количество энергии [12]. Этот алгоритм может быть также использован для оптимального планирования, если критерием оптимизации является линейная комбинация времени выполнения и используемой энергии. Они отметили, что при любом локально-оптимальном плане каждая работа должна выполняться пропорционально количеству неоконченных работ, которые будут задержаны, если эта работа останется невыполненной. Albers and Fujiwara предложили online-версию этого алгоритма планирования и проанализировали его работу в режиме пакетных заданий [13]. Они также сформулировали его эффективную автономную реализацию методом динамического программирования. Bansal, Pruhs и Stein рассмотрели алгоритм, когда мощность процессора, пропорциональна объему незавершенной задачи [14].

2. Минимизация количества энергии либо энергетической эффективности при ограничениях на время отклика или пропускную способность [15].

3. Минимизация взвешенной суммы времени исполнения и затраченной энергии является более общей постановкой. К этой категории относятся алгоритмы интерактивного динамического управления на основе Shortest Remaining Processing Time (SRPT) – кратчайшего оставшегося на обработку времени, либо Highest Density First (HDF – первая работа с наибольшей плотностью) [16].

В одной из первых статей по динамическому управлению питанием [17] был предложен Online-алгоритм энергосберегающего управления устройством с несколькими состояниями: Sleep (сон), Stand-by (ожидание), Idle (бездействие), Active (активность). Было доказано, что стохастические «обучения в режиме Online» имеют преимущества по сравнению с детерминированными стратегиями. Статья [18] развивает этот подход путем одновременного учета двух различных механизмов энергосбережения мобильных устройств. Первый состоит в том, что незагруженная система может быть помещена в состояние сна, что потребует фиксированного количества энергии для возврата в активное состояние. Второй – в управлении скоростью работы процессора. Проблема состоит в том, чтобы запланировать поступающие в систему работы таким образом, чтобы свести к минимуму общее потребление энергии и гарантировать выполнение всех

работ в указанный срок.

Концепция адаптивности в современной теории энергосберегающего управления переносными компьютерами является доминирующей. Собственно адаптивность необходима, прежде всего, для поиска компромисса между экономным расходом остаточной энергии аккумуляторной батареи мобильного устройства и скоростью обработки информации, временем отклика, либо пропускной способностью. При этом необходимо решить 2 взаимозависимые задачи, первая – выбор и установка режима работы процессора является достаточно новой, поскольку возникла только в последние 10-15 лет после появления микропроцессоров, вычислительная мощность которых заведомо превышает средние потребности пользователя мобильного устройства [19]. Вторая – собственно планирование работ (процессов) с учетом ограничений на время их выполнения является традиционной, основы теории работ систем реального времени развиваются с классической работы Лью и Лейланда 1973 года [20].

**Целью работы является** разработка моделей, методов и информационной технологии энергосберегающего планирования задач в условиях автономной работы портативных компьютеров.

## 1. Модель расписания многозадачной системы портативного компьютера в условиях автономной работы

Работа многозадачной системы мобильного компьютерного устройства рассматривается на интервале времени, в течение которого функционирует система:  $t \in [0 \dots T_{\text{sched}}]$ , где  $T_{\text{sched}}$  – максимальный наблюдаемый момент времени. Каждый момент времени определяется как  $t \geq 0$ , где  $t$  – целое число, определяемое подсистемой планирования.

Пусть имеется набор из  $N$  периодических задач  $\tau_i^*$ , подлежащих выполнению:  $\tau = \{\tau_i^*, 1 \leq i \leq N\}$  под управлением многозадачной операционной системы. Каждая  $\tau_i^*$  задача характеризуется следующим набором параметров

$$\tau_i^* = \langle C_i, D_i, T_{o_i}, T_{m_i}, \Delta QoS_i \rangle,$$

где  $C_i$  – время исполнения;

$D_i$  – директивный срок окончания;

$T_{o_i}$  – оптимальный период формирования запросов;

$T_{m_i}$  – максимально допустимый период формирования запросов;

$\Delta QoS_i$  – штраф за потерю качества обслуживания при нарушении директивного срока исполнения

на каждую единицу времени планирования.

Задача  $\tau_i^*$  формирует множество запросов  $\tau_{ij}^*$ , где каждый  $j$ -й запрос при своем появлении помещается в очередь запросов. При этом для всех  $j > 0$ :  $\tau_{ij}^*$  – это время появления запроса,  $s_{ij}^*$  – время начала выполнения запроса (старта), которое в общем случае определяется в ходе диспетчеризации и зависит от метода планирования,  $f_{ij}^*$  – это время завершения выполнения запроса, которое также в общем случае определяется в ходе диспетчеризации.

Если выполнение очередного запроса задачи не может начаться, пока не завершено выполнение всех предыдущих запросов данной задачи, то справедливо соотношение:

$$0 \leq s_{i,j}^* \leq f_{i,j}^* \leq s_{i,j+1}^* \leq f_{i,j+1}^* \quad (4)$$

Для периодической задачи справедливо следующее соотношение:

$$\tau_{i,j}^* = \tau_{i,j-1}^* + T_i = \mu_i + (j-1) \times T_i, \quad (5)$$

где  $\mu_i$  – начальное смещение задачи  $\tau_i^*$ ,

$T_i$  – период формирования запросов периодической задачи.

Пусть  $C_{ij}$  – обозначает продолжительность выполнения  $j$ -го запроса задачи  $\tau_i^*$ . Значение  $C_{ij}$  в общем случае становится известным после момента  $f_{ij}^*$ . Таким образом, продолжительность выполнения запроса  $\tau_{ij}^*$  при отсутствии прерывания вычисляется следующим образом:  $C_{ij} = f_{ij}^* - s_{ij}^*$ .

Во многих случаях сложно заранее определить  $C_{ij}$ , так как количество шагов алгоритма может главным образом зависеть от входных данных и некоторых внутренних переменных. Чтобы заранее гарантировать соблюдение ограничений реального времени обычно используется верхняя оценка возможных значений  $C_{ij}$ .

Пусть  $C_i$  ( $C_i^*$ ) – известная максимальная (минимальная) оценка времени выполнения запроса  $\tau_{ij}^*$ , тогда всегда выполняется соотношение:

$$C_i^* \leq C_{i,j} \leq C_i$$

## 2. Двухкритериальная постановка задачи планирования расписания многозадачной системы мобильного компьютерного устройства

С учетом особенности постановки решаемой задачи составления расписания определим следующие критерии, характерные для систем мягкого ре-

ального времени.

**Критерий 1.** Минимизация среднего значения суммарных штрафов за нарушения директивных сроков на интервале планирования:

$$S_1^* = \arg \min \sum_{i=1}^N \frac{\left( \Delta QoS_i \cdot \sum_{j=1}^{\lfloor T_{\text{sched}}/T_i \rfloor} v_{i,j}(S, V, f) \right)}{\lfloor T_{\text{sched}}/T_i \rfloor}, \quad (6)$$

где  $\lfloor \dots \rfloor$  - операция округления вниз;

$V, f$  - напряжение и частота процессора, аргументы функции энергопотребления (см. (1)).

**Критерий 2.** Минимизация штрафа за сокращение продолжительности работы процессора из-за необходимости обеспечить требуемое качество обслуживания:

$$\langle V^*, f^* \rangle = \arg \min (T_{\text{max}}(V, f) - T_{3\text{max}}(V, f)), \quad (7)$$

где  $T_{\text{max}}(V, f) = E_{\text{ak}}/N_{\text{min}}$  - максимальная продолжительность автономной работы, которая достигается при наличии остаточной энергии аккумулятора  $E_{\text{ak}}$  и минимальной нагрузке  $N_{\text{min}}$ ;

$T_{3\text{max}}(V, f) = E_{\text{ak}}/N_c$  продолжительность автономной работы, которая достигается при текущей нагрузке  $N_c$ .

Функции 6, 7 образуют векторный критерий, который принимает значения в двумерном пространстве  $\mathcal{R}^2$ . Это пространство является *критериальным пространством* или *пространством оценок*, а всякое значение  $y = f(x) = (f_1(x), f_2(x)) \in \mathcal{R}^2$  векторного критерия  $f$  при определенном  $x \in X$  является *векторной оценкой* возможного решения  $x$ . Задача выбора, содержащая множество возможных решений  $X$  и векторный критерий  $f$ , является многокритериальной задачей [21].

Решение  $\bar{x}^* \in X$  называется эффективным решением или оптимальным по Парето решением, если не существует другого решения  $\bar{x} \in X$  среди анализируемых альтернатив, такого, что в множестве  $X$  допустимых альтернативных решений не найдется ни одного другого решения, переход к которому (от  $\bar{x}^*$ ) позволит улучшить показатель хотя бы одного из частных критериев, чтобы при этом не ухудшился показатель второго частного критерия.

Уменьшение напряжения процессора ведет к уменьшению штрафов за потерю продолжительности обслуживания, но при этом возможно увеличение штрафов за потерю качества обслуживания. При

различной глубине разряда аккумулятора (DOD) происходит трансформация кривой предпочтительности на координатной плоскости «Критерий 1»-«Критерий 2» (рис. 1).

Область предпочтений пользователя – это подмножество оптимальных по Парето решений, точки которого являются приемлемыми для пользователя по сочетанию параметров «качество обслуживания» - «продолжительность автономной работы с текущим профилем электропотребления».

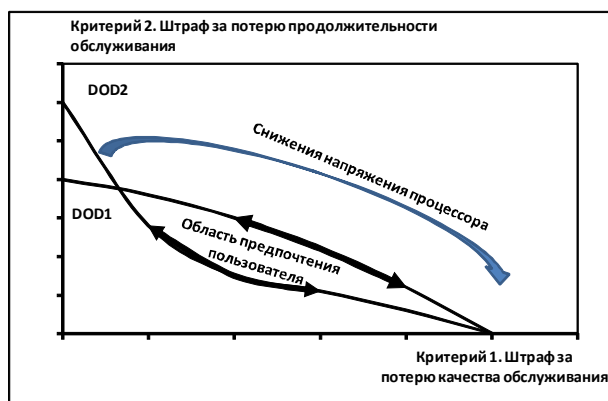


Рис. 1. Парето-оптимальные решения в критериальном пространстве (плоскости): «штраф за нарушения директивных сроков на интервале планирования» - «штраф за сокращение продолжительности работы процессора в условиях автономной работы»

Наиболее очевидно для подсчета остаточной емкости аккумулятора использовать счетчик ампер-часов, который является составной частью многих алгоритмов оценки глубины разряда или остаточной емкости. Он применяется в однокристалльных системах, например, в микросхемах DS2438, DS2780 [22]. Алгоритм определения остаточной емкости по счетчику ампер-часов предполагает вычисление полной емкости аккумулятора при текущих условиях эксплуатации и вычитание из этой величины глубины разряда (затраченной емкости). Основным недостатком при этом – накопление интегральной ошибки при циклическом заряде и разряде батареи.

Для повышения точности работы счетчика ампер-часов применяются различные методы. Так, например, оценить степень заряженности аккумулятора можно при помощи счетчика ампер-часов и самообучающейся системы на основе нечеткой логической модели, которая корректирует показания счетчика при изменении параметров батареи [23].

## Заключение

Планирование задач должно выполняться с учетом характеристик качества обслуживания, фор-

мируемого в зависимости от требований приложений. Различные типы приложений или сервисов требуют различные скорости передачи данных, сетевых задержек, уровни надежности и безопасности. Насыщенные данными приложения, такие как потоковое видео, будут работать лучше, если пропускная способность будет высокой. Приложениям, которые работают в реальном времени, необходима минимальная задержка, прочие приложения менее чувствительны к задержкам.

Разработанная модель расписания многозадачной системы портативного компьютера в условиях автономной работы основана на предположении, что система относится к классу систем мягкого реального времени. Нарушение директивных сроков в этом случае приводит к снижению качества обслуживания, особенно заметным такое снижение качества проявляется в голосовой телефонии, видео телефонии, видео-конференциях, играх в режиме реального времени и т.д. Синтез расписания многозадачной системы портативного компьютера в условиях автономной работы выполняется путем решения двухкритериальной оптимизационной задачи с выделением множества Парето-оптимальных решений. К числу критериев относятся минимум штрафов за снижение качества обслуживания и потерю продолжительности автономной работы с текущим профилем электропотребления. Сужение множества Парето-оптимальных решений возможно на основе учета предпочтений пользователя.

Дальнейшим развитием работы является разработка метода адаптивного управления процессом планирования задач в условиях автономной работы мобильного устройства с учетом двухкритериальной постановки задачи оптимизации.

## Литература

1. *Energy consumption of consumer electronics in u.s. homes in 2013. Final report to the consumer electronics association [Electronic resource]. – Access mode: [https://www.google.com.ua/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CDEQFjAAahUKEwi\\_3tbBtI\\_JAhXon3IKHYpwBWM&url=http%3A%2F%2Fwww.ce.org%2FCorporateSite%2Fmedia%2Fenvironment%2FEnergy-Consumption-of-Consumer-Electronics.pdf&usq=AFQjCNF\\_oP4zVgGm7BY-pwZEzC5wGEwivQ&sig2=uvw5dRjzZuoZL16McuGfKQ](https://www.google.com.ua/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CDEQFjAAahUKEwi_3tbBtI_JAhXon3IKHYpwBWM&url=http%3A%2F%2Fwww.ce.org%2FCorporateSite%2Fmedia%2Fenvironment%2FEnergy-Consumption-of-Consumer-Electronics.pdf&usq=AFQjCNF_oP4zVgGm7BY-pwZEzC5wGEwivQ&sig2=uvw5dRjzZuoZL16McuGfKQ). - 02.11.2015.*
2. *Enhanced Intel SpeedStep Technology for the Intel Pentium M Processor White Paper (PDF). Intel (March 2004) [Electronic resource]. – Access mode: [/ftp://download.intel.com/design/network/papers/30117401.pdf](http://download.intel.com/design/network/papers/30117401.pdf). – 02.11.2015.*

3. Snowdon, D. *Power Management and Dynamic Voltage Scaling: Myths and Facts [Text]* / D. Snowdon, S. Ruocco, G. Heiser // *Proceedings of the National ICT Australia and School of Computer Science and Engineering University of NSW.* – Australia, 2005. – P. 15-19.
4. Mudge, T. *Power: A first-class architectural design constraint [Text]* / T. Mudge // *Computer.* – 2001. – № 34(4). – P. 52-58.
5. Bansal, N. *Speed scaling for weighted flow time [Text]* / N. Bansal, K. Pruhs, C. Stein // *Proceeding SODA '07 Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms.* – 2007. – P. 805-813.
6. Bansal, N. *Speed scaling to manage energy and temperature [Text]* / N. Bansal, T. Kimbrel, P. Kirk // *Journal of the ACM, Article 3.* – March 2007. – № 54(1). – P. 1-39.
7. Bansal, N. *Speed scaling with an arbitrary power function [Text]* / N. Bansal, H. Chan, K. Pruhs // *Proceeding SODA '09 Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms.* – 2009. – P. 693-701.
8. Albers, S. *Energy-efficient algorithms for flow time minimization [Text]* / S. Albers, H. Fujiwara // *ACM Transactions on Algorithms (TALG).* – November 2007. – Vol. 3, Iss. 4. – P. 32-44.
9. *Energy efficient online deadline scheduling [Text]* / Ho-Leung Chan, Wun-Tat Chan, Tak-Wah Lam, Lap-Kei Lee, Kin-Sum Mak, Prudence W. H. Wong // *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms.* – 2007. – P. 795-804.
10. Граничин, О. *Алгоритмы оптимизации энергопотребления в мобильных устройствах [Текст]* / О. Н. Граничин, В. Е. Краснощеков ; под ред. д-ра техн. наук, проф. Ю. И. Нечаева // *Сборник статей «Современные проблемы нейроиформатики».* – 2007. – Часть 3, Книга 23. – С. 18-25.
11. *Dynamic voltage scaling and power management for portable systems [Text]* / T. Simunic, L. Benini, A. Acquaviva, P. Glynn, G. De Micheli // *Design Automation Conference.* – 2001. – P. 524-529.
12. Pruhs, K. *Getting the best response for your erg [Text]* / Kirk Pruhs, Patchrawat Uthaisombut, Gerhard Woeginger // *ACM Transactions on Algorithms (TALG).* – June 2008. – Vol. 4, Iss. 3. – P. 121-142.
13. Albers, S. *Energy-efficient algorithms for flow time minimization [Text]* / S. Albers, H. Fujiwara // *In Lecture Notes in Computer Science (STACS).* – 2006. – Vol. 3884. – P. 621 – 633.
14. Bansal, N. *Speed scaling for weighted flow time [Text]* / N. Bansal, K. Pruhs, C. Stein // *In SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms.* – 2007. – P.805–813.
15. *Online Deadline Scheduling with Bounded Energy Efficiency [Text]* / Joseph Wun-Tat Chan, Tak-Wah Lam, Kin-Sum Mak, Prudence W. H. Wong // *Theory and Applications of Models of Computation. Lecture Notes in Computer Science.* – 2007. – Vol. 4484. – P. 416-427.
16. Lachlan, L. H. Andrew. *Optimal speed scaling under arbitrary power functions [Text]* / Andrew L. H. Lachlan, Adam Wierman, Ao Tang // *ACM SIGMETRICS Performance Evaluation Review archive.* – September 2009. – Vol. 37, Iss 2. – P. 39-41.
17. Irani, S. *Online strategies for dynamic power management in systems with multiple power-saving states [Text]* / S. Irani, S. Shukla, R. Gupta // *ACM Transactions on Embedded Computing Systems (TECS).* – August 2003. – Vol. 2, Iss. 3. – P. 325-346.
18. Irani, S. *Algorithms for power savings ACM [Text]* / S. Irani, S. Shukla, R. Gupta // *Transactions on Algorithms (TALG).* – November 2007. – Vol. 3, Iss. 4. – P. 34-57.
19. Мусиенко, М. П. *Оптимальное распараллеливание задач диспетчеризации в распределенных энергоограниченных компьютерных системах на базе MCU FPGA модулей [Электронный ресурс]* / М. П. Мусиенко, В. Ю. Савинов, Я. М. Крайнык // *Вектор науки ТГУ.* – 2014. – № 1 (27). – Режим доступа: <http://cyberleninka.ru/article/n/optimalnoe-rasparalleliwanie-zadach-dispetcherizatsii-v-raspredeleennyh-energoogranichennyh-kompyuternyh-sistemah-na-baze-msu-frga>. – 28.09.2015.
20. Liu, C. L. *Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment [Text]* / C. L. Liu, James W. Layland. // *Journal of the ACM.* – 1973. – № 20. – P. 46 – 61.
21. Емельянов, С. В. *Многокритериальные методы принятия решений [Текст]* / С. В. Емельянов, О. Й. Ларичев. – М. : Знание, 1985. – 32 с.
22. Хандорин, М. М. *Оценка остаточной емкости литий-ионного аккумулятора на основе эталонной модели [Текст]* / М. М. Хандорин, В. Г. Букреев // *Вестник СубГАН.* – 2014. – № 1(53). – С. 94-97.
23. *Determination of state-of-charge and state-of-health of batteries by fuzzy logic methodology [Text]* / A. J. Salkind, C. Fennie, P. Singh, T. Atwater, D. E. Reisner // *Journal of Power Sources.* – 1999. – Vol. 80, No. 1-2. – P. 293–300.

Поступила в редакцию 23.10.2015, рассмотрена на редколлегии 18.11.2015

## ПРИНЦИП АДАПТИВНОГО ПЛАНУВАННЯ ЗАВДАНЬ В УМОВАХ АВТОНОМНОЇ РОБОТИ ПОРТАТИВНОГО КОМП'ЮТЕРА

*С. С. Аль-Кхшаб*

Розглянуто принципи управління продуктивністю сучасних процесорів, в основі яких енергозберігаюча технологія динамічного масштабування напруги і частоти процесора. Показано, що для ефективного застосування динамічного масштабування необхідний синтез розкладу завдань з урахуванням обмежень м'якого реального часу. Синтез розкладу заснований на вирішенні двукритеріальної оптимізаційної задачі з виділенням множини Парето-оптимальних рішень. До критеріїв належать мінімуми штрафів за зниження якості обслуговування і втрату тривалості автономної роботи з поточним профілем електроспоживання. Звуження множини Парето-оптимальних рішень можливе на основі врахування уподобань користувача.

**Ключові слова:** адаптивність, динамічна зміна напруги і частоти процесора, м'який реальний час, Парето-оптимальне рішення, процесор, оптимізація, енергозбереження.

## PRINCIPLES OF THE ADAPTIVE TASKS PLANNING IN THE BATTERY LIFE OF PORTABLE COMPUTER'S

*S. S. Al-Khshab*

The principles of performance management of modern processors, which are based energy saving technology dynamically changed the voltage and frequency of the processor, are presented. It is shown that effective energy-saving use leads to the necessity of synthesis schedules tasks based soft real-time constraints. Synthesis of schedules are based on the two-criteria optimization problem solutions with the release of the Pareto-optimal solutions. The criteria include minimum penalties for the decline in the quality of service and the loss of battery life with the current power consumption profile.

**Keywords:** adaptive, dynamic processor frequency and voltage scaling, soft real-time, Pareto optimal solution, processor, optimization, energy saving.

**Аль-Кхшаб Синан Самир** – аспірант каф. інженерії програмного забезпечення, Национальный аэрокосмический университет им. Н. Е. Жуковского «Харьковский авиационный институт», Харьков, Украина, e-mail: [sinan\\_alkhshab@mail.ru](mailto:sinan_alkhshab@mail.ru).