Energy Optimization of Stochastic Applications with Statistical Guarantees of Deadline and Reliability

Xiong Pan[1], Wei Jiang[1], Ke Jiang[2], Liang Wen[1], and Qi Dong[1]

[1]School of Information and Software Engineering
University of Electronic Science and Technology of China

[2]Department of Computer and Information Science, Linköping University, Sweden

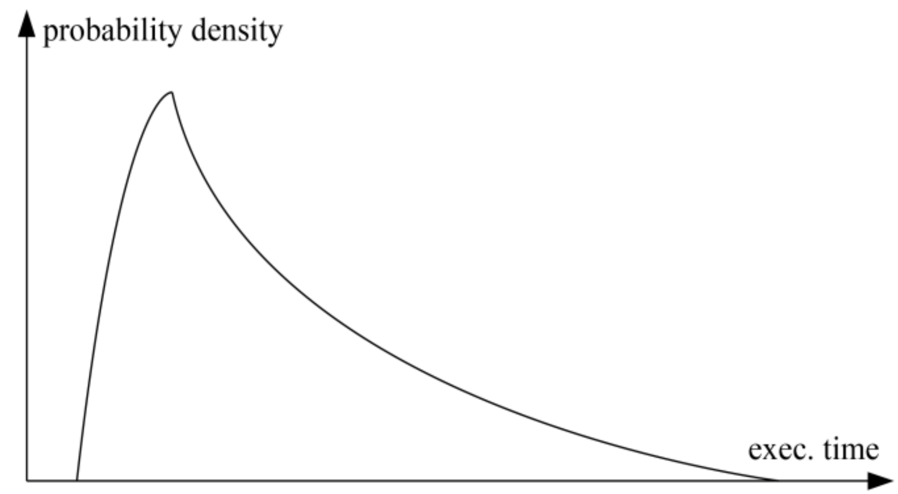
Outline of this paper

- Introduction
- System model
- System problem formulation
- Proposed approaches based on dynamic programming
- Experimental results
- Conclusions

Introduction

Paper focus on:

- Optimizing energy consumption of stochastic tasks;
- Take into account the probability distribution of task's execution time;



- The application has reliability and deadline constraints;
- Running on DVFS-enabled embedded systems.

System Model

Stochastic real-time task model:

Scenario and Assumption

- An embedded server to provide real-time services for multiple users in a common period.
- All service tasks are released in each period, and finished before their respective deadlines.

We use (D_i, C_i, PDF_i) to characterize the task *i*.

- D_i: relative deadline;
- C_i: stochastic execution time;
- PDF_i: probability density function of C_i.

System Model

DVFS-enabled energy and reliability model:

- Expected energy consumption of task i:

$$\overline{En_i} = (Pow_i^{dyn} + Pow_i^{leak}) \cdot \overline{C_i}$$

- Reliability of task i:

$$Re_i = exp(-\frac{\lambda(\phi_i) \cdot C_i}{f_i})$$

- Expected logarithm of the reliability of the whole application can be calculated as:

$$\overline{\ln(ORe)} = \sum_{i=1}^{n} (-\frac{\lambda(\phi_i) \cdot \overline{C_i}}{f_i})$$

System Model

DVFS-enabled energy and reliability model

We need $\ln\left(\overline{ORe}\right)$ or $\overline{ORe}$ for calculating the reliability, but we only have go $\overline{\ln\left(ORe\right)}$, so formulating the following theorem:

Theorem 1 *The natural logarithm of expected system reliability can be expressed as* $\ln\left(\overline{ORe}\right) = \overline{\ln\left(ORe\right)}$*, approximately.*

The theorem can be proved by Taylor expansion and Taylor Mean Value Theorem.

System Model

Modeling the Probability of No-Deadline Violation

To model the deadline constraint of soft real-time applications, we introduce a metric, Probability of No-Deadline Violation (PNDV), indicating that the deadline violation for each task can only be tolerated by certain probability.

- Pb_i^h denotes the probability of the situation when the actual execution time of τ_i is h;
- OPb_i^h denotes the probability that the overall execution duration of the previous i tasks is h;
- Te_i denotes the sum of the execution durations of the previous i tasks.

System Model

Modeling the Probability of No-Deadline Violation

Sum of the execution durations of the previous i tasks:

$$Te_i = \sum_{j=1}^{i} C_j.$$

Given the probability distribution of Te_{i-1}, Probability distribution of Te_i can be calculated easily by:

$$OPb_i^h = \sum_{x+y=h} (OPb_{i-1}^x \cdot Pb_i^y)$$

At last, *PNDV* of task *i* can be expressed as follows:

$$PNDV_i = \sum_{h=1}^{D_i} OPb_i^h$$

System Problem Formulation

The problem we address is described as follows. Given a task set with n tasks $\tau_1, \tau_2, \ldots, \tau_n$, with corresponding deadline, execution time PDF and m possible levels of processor frequency, we are interested to determine a DVFS solution, i.e. S = (*f1, f2, … , fn*), such that expected system energy for a period of the tasks is minimized and the constraints of the reliability (ORe^B) and deadlines ($PNDV^B$) are met.

The optimization objective can be formulated as:

$$Minimize\ \ OEn = \sum_{i=1}^{n} (Pow_i^{dyn} + Pow_i^{leak}) \cdot Ept(C_i)$$

System Problem formulation

Soft deadline constraint of individual task can be expressed:

$$PNDV_i \geq PNDV_i^B, 1 \leq i \leq n$$

Reliability constraint can be expressed:

$$\sum_{i=1}^{n}\left(-\frac{\lambda(\phi_i)\cdot\overline{C_i}}{f_i}\right) \geq \ln(ORe^B), f_i \in \{F_1, F_2, \cdots, F_m\}$$

{F1,F2, … ,Fm} are the normalized discrete frequency options for the DVFS-enabled processor.

Proposed Approaches

DPO approach(Dynamic Programming based Optimization algorithm)

It is a Multidimensional Multiple-Choice Knapsack Problem.

The state transition equation can expressed as following:

$$\Phi(i,\delta,k) = max^k\{\Phi(i,\delta-1,l),$$
$$max\{\Phi(i-1,\delta-\overline{En_i^j},l)+\overline{\ln Re_i^j}\}\}$$
$$1 \le j \le m, 1 \le l \le k$$

- $\Phi(i,\delta,k)$: expected natural logarithm of the k-th largest reliability of the previous i tasks, whose expected total energy is limited to no more than δ.
- En_i^j and Re_i^j: energy and reliability of τi using the j-th frequency level.

Proposed Approaches

FBDP approach

(Fast Bi-search algorithm based on Dynamic Programming)

- Due to high runtime complexity of DPO, we propose FBDP .
- FBDP is to find a solution satisfying both reliability and deadline constraints with the energy cost at most $(1+\beta)OEn^*$, where OEn^* denotes the expected energy of our optimal problem.
- Achieving this, a Bi-searcher will be first constructed such that a bi-search on whether $OEn^* \geq x$ for any x can be found in polynomial time without knowing the value of OEn^*. We can use Bi-searcher to bi-narrow the range of OEn^*.
- After constructing such a Bi-searcher, the range of OEn^*, which is narrow enough, can be obtained by performing a fast bi-search between the lower and upper bounds of OEn^*.

Proposed Approaches

FBDP approach

Construction of Bi-searcher:

Scale En_i^j to $\lfloor En_i^j \cdot n/(x\beta) \rfloor$ and reliability and deadline constraints are unchanged. The upper bound of expected energy cost is set to be *n/β*, after executing DPO and we can determine the range of *OEn** based on the following 2 cases:

Case 1: Found a solution satisfying both reliability and deadline constraints. Then *OEn* ≤ (1+β)x*.

Case 2: No solution is found that satisfied both reliability and deadline constraints. Then *OEn*> x.*

Proposed Approaches

FBDP approach

After each bi-search, we can make corresponding operations according to the previous two cases:

Case 1: $OEn^* \leq (1+\beta)x$. Set new upper bound to $(1+\beta)x = \sqrt{(OEn_l^* \cdot OEn_u^*)/(1+\beta)}$ and keep lower bound unchanged.

Case 2: $OEn^* > x$. Set new lower bound to $\sqrt{(OEn_l^* \cdot OEn_u^*)/(1+\beta)}$ and keep upper bound unchanged.

It can be proved that the ratio of upper bound to lower bound is no more than $2(1+\beta)$ after $\log(\log Q_0)$ bi-search operations, and obtain the near-optimal solution with the energy cost at most $(1+\beta)OEn^*$ without violating the reliability and deadline constraints.

Experimental Results

Parameters on tasks in all the experiments are generated randomly:

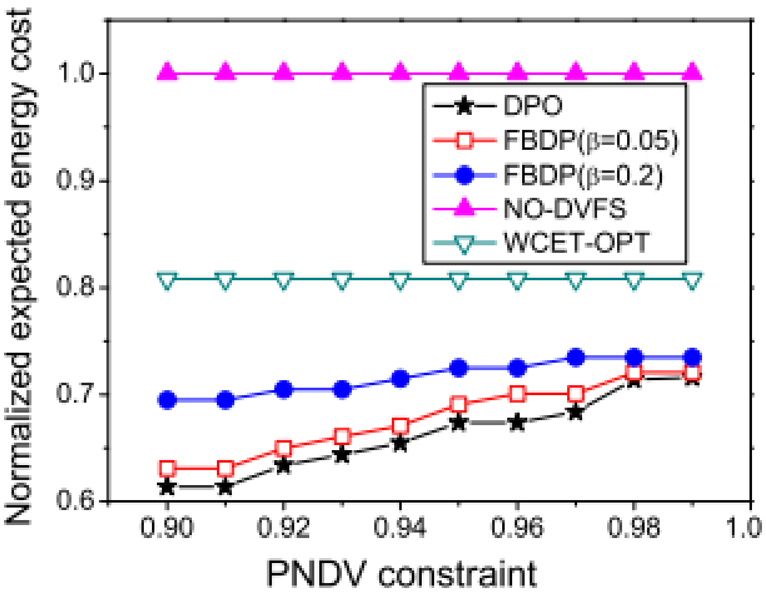
- Deadline -- [30 ms, 2000 ms]
- WCET -- [20 ms, 80 ms]
- PDF -- [Gaussian Distribution, Uniform Distribution or Exponential Distribution]
- Expected values of tasks' execution durations -- [5, 30]
- variances of tasks' execution durations -- [0.2, 2]

Comparing algorithms

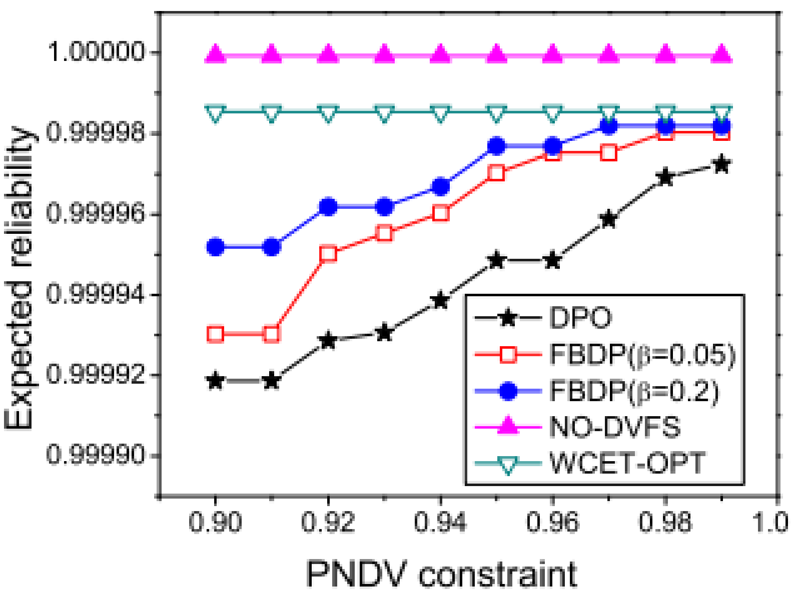
- WCET-OPT -- takes the WCET as the actual execution time of each task;
- NO-DVFS -- ignores DVFS capability of the processor;
- DPO -- Dynamic Programming based Optimization algorithm;
- FBDP -- Fast Bi-search algorithm based on Dynamic Programming.

Experimental Results

Impact of PNDV constrain:



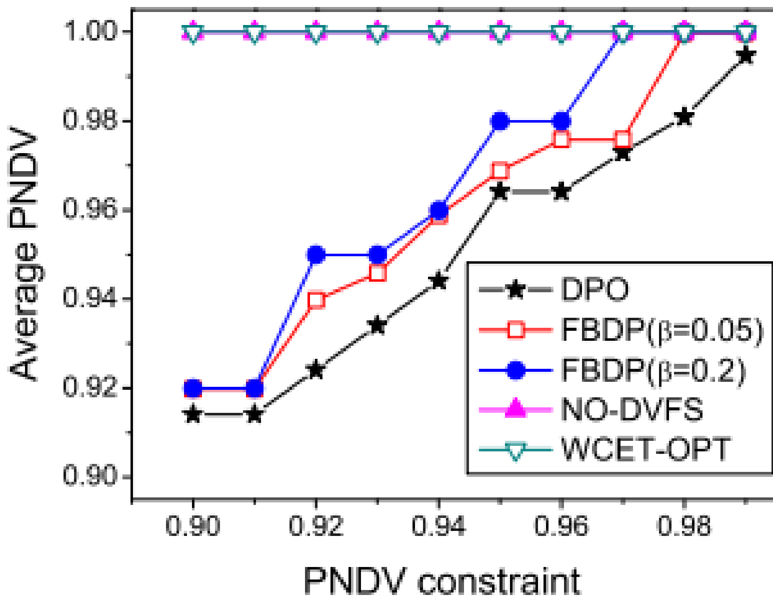
(a)



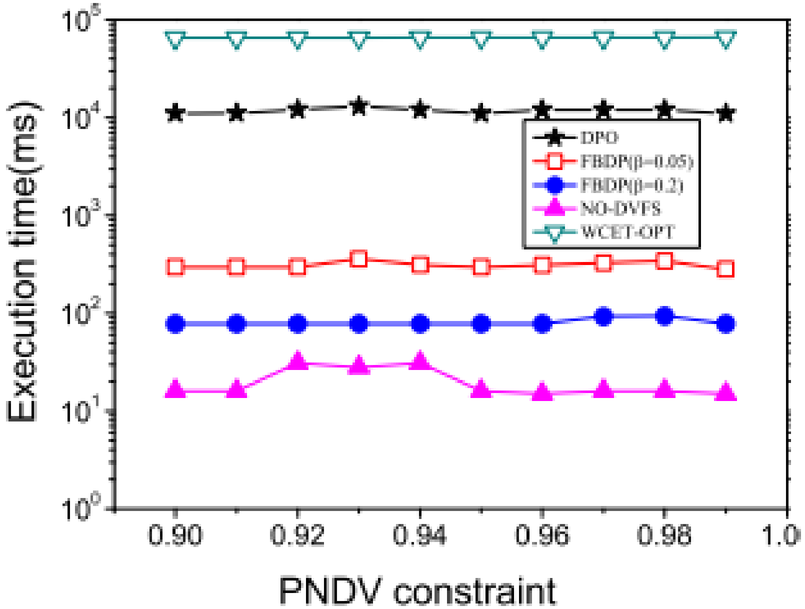
(b)

Experimental Results

Impact of PNDV constrain:



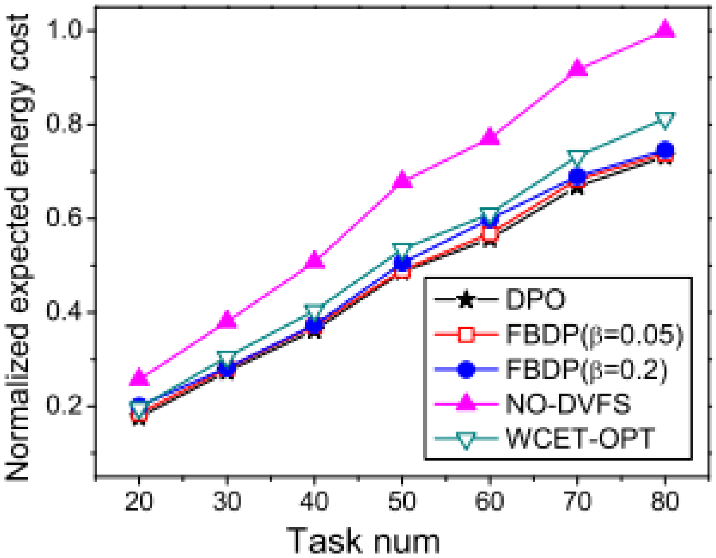
(c)



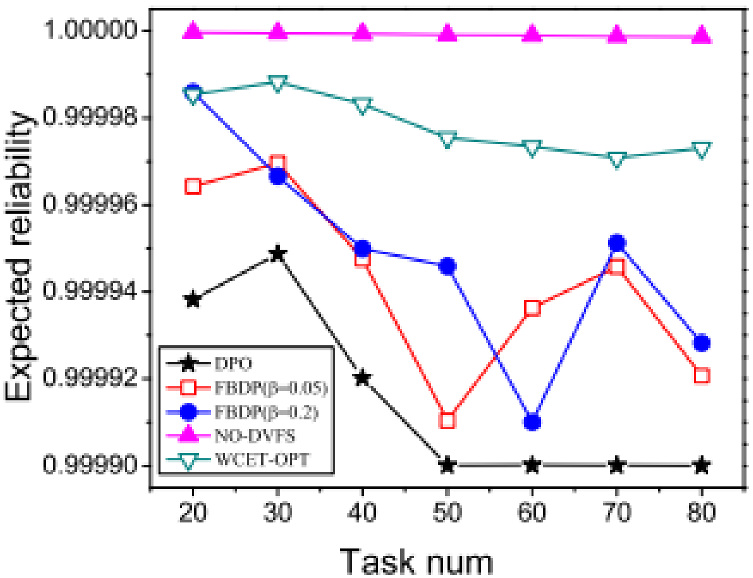
(d)

Experimental Results

Impact of task number:



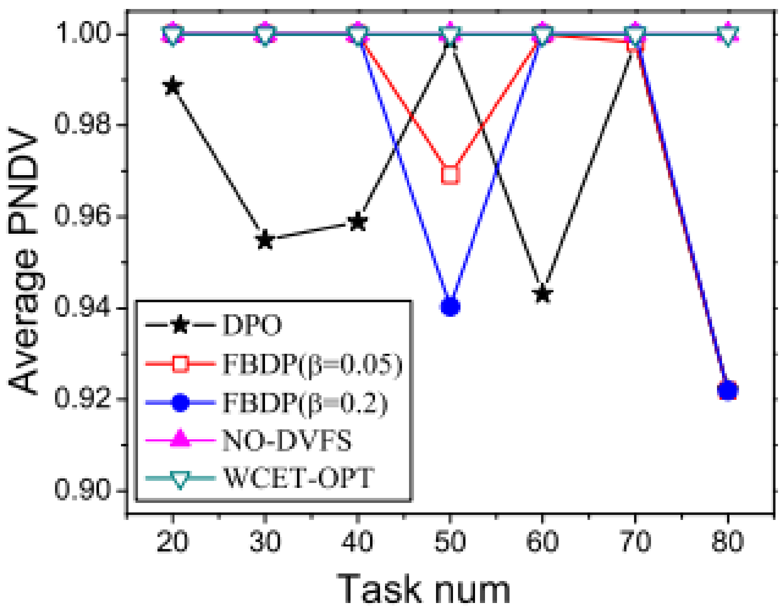
(a)

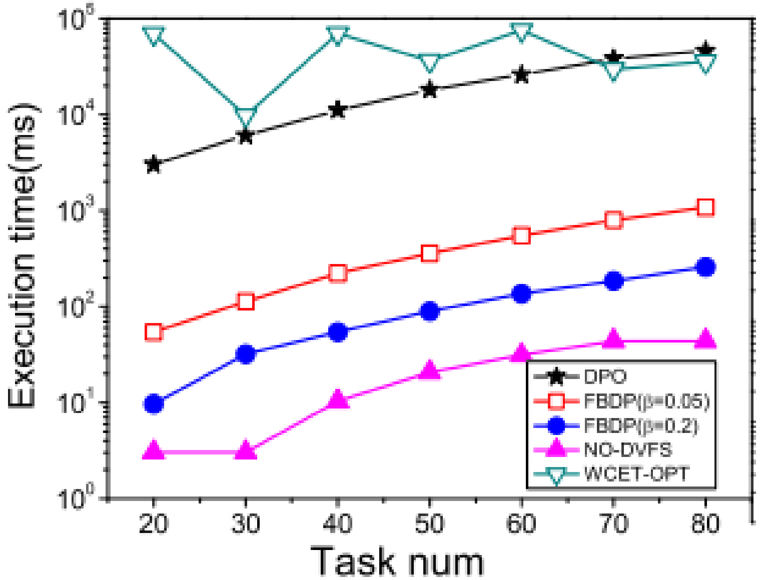


(b)

Experimental Results

Impact of task number:





Experimental Results

A real-life case :

We consider an embedded server like satellite server or radar server, which has to provide services for 30 users within a common period of 3000 ms. The task for each user all arrives at the same time and has to be finished before next period.

PNDV Constraint	DPO				FBDP(β=0.05)				WCET-OPT		
	OEn	*NNRV*	*NNDV*	*Time(ms)*	*OEn*	*NNRV*	*NNDV*	*Time(ms)*	*OEn*	*NNRV*	*Time(ms)*
0.91	0.71536	9926	289456	6078	0.74269	9930	290386	148	1	10000	35510
0.93	0.71536	9926	289456	8063	0.75317	9951	291323	197	1	10000	35712
0.95	0.76451	9949	292528	6047	0.78376	9973	295008	149	1	10000	35946
0.97	0.84789	9971	295984	7063	0.87254	9979	296121	163	1	10000	36025
0.99	0.92214	9987	298416	6062	0.93998	9989	299367	141	1	10000	36213

Conclusions

- Consider PDFs of tasks' execution into the design problem;
- DVFS is used for energy saving;
- Design FBDP which can achieve approximated optimal solution;
- Experiment results illustrated the effectiveness of proposed technique;
- used Monte Carlo simulation to simulate the stochastic tasks for verifying the proposed approaches.

The end

That is all my presentation
Thanks for your time