

An Efficient Decoupling Capacitance Optimization Using Piecewise Polynomial Models

Xiaoyi Wang*, Yici Cai*, Sheldon X.-D. Tan[†], Xianlong Hong*, Jacob Relles[†]

*Department of Computer Science and Technology, Tsinghua University, Beijing, China, 100084

[†]Department of Electrical Engineering, University of California, Riverside, CA 92521

Abstract—This paper proposes an efficient decoupling (decaps) capacitance optimization algorithm to reduce the voltage noise of on-chip power grid networks. The new method is based on the efficient charge formulation of the decap allocation problem. But different from the existing work [12], the new method applies the more accurate piecewise polynomial micromodels to estimate the voltage noises during the linear programming process. The resulting method overcomes the over-estimation problem, which plagues the existing method. The proposed method has the best of two worlds: it has the efficiency of the charge-based methods and the accuracy of the sensitivity-based methods. Experimental results demonstrate that the proposed method leads to the decap values similar to that of the sensitivity-based methods, which give the best reported results and are much better than the existing charge-based method, and at the same time, it enjoys the similar efficiency of the charge-based method.

I. INTRODUCTION

With increasing integration density and roaring clock frequency, reliable on-chip power supply becomes a challenging task for VLSI chip design. Excessive voltage variations, including voltage drop and ground bounce, would have an adverse impact on chip performance and reliability, since they not only degrade the already tight noise margin in today's VLSI circuits, but also increase gate delay, cause false logic switching, and sometimes even lead to logic failure. In practice, most designs now require voltage drop to be confined to certain percentage of the nominal supply voltage [1]. Adding decoupling capacitance (decaps) has been proved to be the most efficient way to restrict the voltage drops in on-chip power grid networks. However excessive decaps will cause more leakage currents and reduce the reliability of the chips. As a result, decap should be added in an economical way to optimally balance its noise reduction functions and adverse effects.

Budgeting decap in an area-efficient way, however, is a difficult task due to prohibitive analysis costs of power/grid (P/G) networks with millions of nodes and extracted on-chip and off-chip RLC components in modern VLSI design. Mathematically, optimal decap allocation is a nonlinear optimization problem [9]. Existing approaches can be roughly classified into two categories: the sensitivity based methods and the charge-based methods.

Sensitivity-based approaches [2], [4]–[6], [8], [10] use sensitivity information to guide the optimization processes. To

compute the sensitivity, transient simulations of the whole P/G networks have to be carried out at every optimization step. As a result, those methods are very expensive and difficult to scale for large circuits. To mitigate the problem, multigrid concept was proposed in [10], where the sensitivity-based optimization is performed on a size-reduced mesh. The method, however, is limited to the regular-mesh or regular-mesh-like P/G structures. In [6], the whole circuit is partitioned into a number of sub-circuits based on the local effect of adding decap to reduce IR drops, and each of them is optimized individually. In [2], leakage current impacts of decaps were explicitly considered in sensitivity-based framework. Recently the partitioning is done more efficiently using random walk concept [5]. Sensitivity based method has the advantage over charge-based methods in that they can deliver more accurate and smaller decap values as they directly monitor the voltage drops during the optimization processes.

The second methods are operating on the charges or the transformed voltage drops (the voltage-time integral), which can lead to more simple problem formulations [3], [11], [12]. Those methods perform the approximate integration on the differential equations. As a result, no transient simulations are required during the optimization and the methods are much more efficient than the sensitivity-based methods. However, they generally suffer over-estimation issue as the approximation can be quite rough as we will demonstrate in this paper.

In this paper, we propose an efficient charge-based decap optimization method, which has the best of two worlds: it has the efficiency of the charge-based methods and the accuracy of the sensitivity-based methods. The new method is based on more accurate modeling of the voltage-time integral constraints, which is critical for the accurate estimation of decap during the optimization. We propose a more accurate piecewise polynomial model to approximate the voltage-time integral instead of the simple linear model as in the existing method [12]. We show that the existing approach can lead to significant over estimation of the decap values. Experimental results show that the new approach delivers the much smaller decap values, similar to the sensitivity-based method, than the latest charge-based method with similar computational costs of charge-based methods.

II. PROBLEM FORMULATION

The purpose of decap optimization is to use minimum decap to avoid the excessive voltage drop. At the same time, the

This work is funded in part by National Natural Science Foundation of China (NSFC) grants under No. 60828008 and No. 60776026 and in part by National Science Foundation (NSF) grant under No. OISE-0623038.

maximum single capacitance is bounded due to the physical constraint. Then we can formulate the decap optimization problem as following:

$$\begin{aligned} \min \quad & \sum_i c_i \\ \text{subject to} \quad & C \cdot \frac{d\mathbf{V}}{dt} + G \cdot \mathbf{V} = \mathbf{J} \\ & \mathbf{V} \geq V_{\text{thre}} \\ & c_{\text{int},i} \leq c_i \leq c_{\text{max},i} \end{aligned} \quad (1)$$

$$(2)$$

where c_i is the capacitance, which includes both intrinsic and decoupling capacitances and $c_{\text{int},i}$ is the intrinsic capacitance at node i . $C = \text{diag}(c_1, \dots, c_n)$, $\mathbf{V}$ is the transient voltage waveform vector, V_{thre} is the threshold voltage, $c_{\text{max},i}$ is the maximum capacitance allowed at node i , G is the conductance matrix obtained by modified nodal analysis (MNA) and $\mathbf{J}$ is the current source vector. In our formulation, we ignore the inductances as inductances are still less important at the chip level [7].

III. REVIEW OF SLP-BASED DECAP BUDGET METHOD

There are four steps in the successive linear programming (SLP) based decap optimization method [12] as described in algorithm 1.

- 1 Run the transient simulation of power/ground network once with the intrinsic capacitances and placed decap ;
- 2 Find out the violation regions, (i.e. the regions where the voltage drop constraints are violated) and the violation time windows (i.e. the time windows during which violations occur) . Also Decide the sampling nodes, i.e. the nodes to which the decap will be attached. If no violation is found, the optimization ends ;
- 3 For each violation region, determine the optimal amount of decap by linear programming or sequential linear programming ;
- 4 Distribute the decap of the sampling nodes into the respective sampling regions. Go back to step 1 until no violation is found;

Algorithm 1: The flow of decap optimization in [12].

In the critical third step of algorithm 1, the decap optimization problem (1) is formulated as a linear programming problem.

The essential idea of this formulation is to transform the MNA equations and voltage drop constraints (2) to linear constraints by integration and approximation of voltage waveforms. First, the MNA equations (2) are transformed to “charge transfer equations” by integration and macromodeling. Suppose we have already found out the violation regions and sampling nodes by transient simulation and the violation time window is determined as $[t_0, t_1]$ (please refer to [12] for more details.). Then the MNA equation (2) is partitioned as follows:

$$\begin{pmatrix} G_{11} & G_{12} \\ G_{12}^T & G_{22} \end{pmatrix} \cdot \begin{pmatrix} \int_{t_0}^{t_1} \mathbf{V}_1 dt \\ \int_{t_0}^{t_1} \mathbf{V}_2 dt \end{pmatrix} = \begin{pmatrix} \int_{t_0}^{t_1} \mathbf{J}_1 dt \\ \int_{t_0}^{t_1} \mathbf{J}_2 dt + \int_{t_0}^{t_1} \mathbf{I} dt \end{pmatrix}$$

where $\mathbf{V}_2$ is the voltages of the sampled violating nodes and $\mathbf{I}$ is the current supplied by decap. Then the charge transfer equations are formulated by macromodeling :

$$\mathbf{Q} = \mathbf{A} \cdot \mathbf{W} + \mathbf{B} \quad (3)$$

where $\mathbf{Q} = \int_{t_0}^{t_1} \mathbf{I} dt$ is the charge transferred from decap, $\mathbf{A} = G_{22} - G_{12}^T G_{11}^{-1} G_{12}$ is the admittance matrix, $\mathbf{W}$

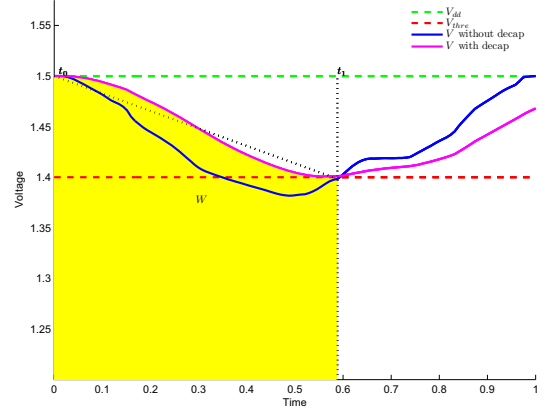


Fig. 1. Voltage Waveforms without & with decap

is $\int_{t_0}^{t_1} \mathbf{V}_2 dt$ and $\mathbf{B} = (G_{12}^T G_{11}^{-1} \int_{t_0}^{t_1} \mathbf{J}_1 dt - \int_{t_0}^{t_1} \mathbf{J}_2 dt)$ is the equivalent charge drawn by current sources. Second, the voltage constraints (2) are also transformed to linear constraints on voltage-time integral $\mathbf{W}$. As shown in Figure 1, the voltage waveform should be above the V_{thre} after adding decap. Correspondingly, the integral of voltage waveform ($\mathbf{W}$) should be no less than the shaded area. However, the shaded area is unknown until the decap is added, [12] suggests using the linear approximation of this area. Hence, the transient voltage constraint (2) is re-formulated as:

$$\mathbf{W} \geq \frac{1}{2} (\mathbf{V}(t_0) + V_{\text{thre}})(t_1 - t_0)$$

Since all the constraints in problem (1) have been transferred to linear constraints, we end up with a linear programming problem:

$$\min \quad \sum_{i \in \mathbf{S}} c_i \quad (4)$$

$$s.t. \quad \mathbf{M}' \circ \mathbf{C} \geq \mathbf{A} \cdot \mathbf{W} + \mathbf{B}$$

$$\mathbf{W} \geq \mathbf{L} \quad (5)$$

$$c_{\text{int},i} \leq c_i \leq c_{\text{max},i}$$

where $\mathbf{C} = (c_1 \ c_2 \ \dots \ c_m)^T$,

$$\mathbf{M}' = \begin{pmatrix} \mathbf{V}_1(t_0) - \mathbf{V}_1(t_1) \\ \mathbf{V}_2(t_0) - \mathbf{V}_2(t_1) \\ \vdots \\ \mathbf{V}_m(t_0) - \mathbf{V}_m(t_1) \end{pmatrix}, \mathbf{L} = \begin{pmatrix} (V_{\text{thre}} + \mathbf{V}_1(t_0)) \cdot T \\ (V_{\text{thre}} + \mathbf{V}_2(t_0)) \cdot T \\ \vdots \\ (V_{\text{thre}} + \mathbf{V}_m(t_0)) \cdot T \end{pmatrix}, \quad (6)$$

$\mathbf{S}$ is the set of sampling nodes, $m = |\mathbf{S}|$, $T = (t_1 - t_0)/2$ and the $\circ$ operation is the entry-wise product.

There are, however, two inaccurate variables in such linear programming formulation. The first one is $\mathbf{V}(t_1)$. Ideally, $\mathbf{V}$ should reach the threshold voltage at t_1 , i.e. $\mathbf{V}(t_1) = V_{\text{thre}}$. However, $\mathbf{V}(t_1)$ could also be above V_{thre} practically. As a result, [12] proposed a successive linear programming (SLP) to mitigate this problem. The second non-accurate variable is $\mathbf{L}$, the lower bound of the voltage-time integral. As illustrated in Figure 1, the voltage waveform after adding decap is approximated by a straight line. So $\mathbf{L}$ is the area of trapezoid,

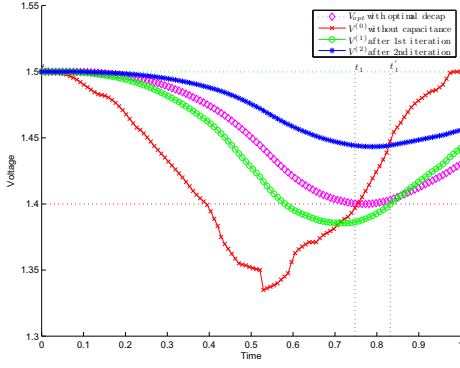


Fig. 4. Simple iteration of the SLP method leads to decap over-estimation.

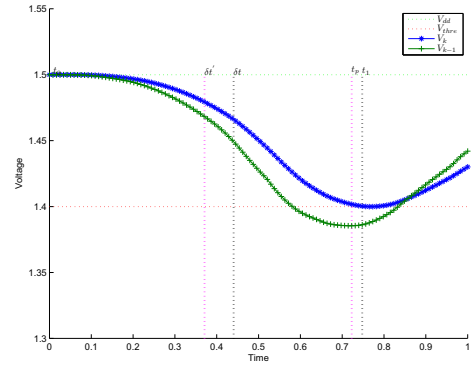


Fig. 5. Piecewise polynomial modeling of voltage waveform integration.

i.e. $V^{(2)}(t'_1) > V_{\text{thre}}$, which is also shown in Figure 4. Actually, we can see this fact from the MNA equation $C \cdot \frac{dV^{(2)}}{dt} + G \cdot V^{(2)} = J$. Since $\frac{dV^{(2)}}{dt}(t'_1) \approx 0$ (because t'_1 is minimum voltage time point as mentioned before), so we know $G \cdot V^{(2)}(t'_1) \approx I(t'_1)$. While we also know $G \cdot V^{(0)} = I$ (the MNA with $C = 0$), we conclude that $V^{(2)}(t'_1) \approx V^{(0)}(t'_1) > V^{(0)}(t_1) = V_{\text{thre}}$. This deduction is also proved by the waveforms in Figure 4.

Therefore, the simple iteration of SLP pulls the voltage much higher than necessary and consequently costs more decap than necessary. Table II shows the overall decap amount during the iteration of SLP. C_{opt} is the decap amount obtained by sensitivity based decap optimization and C is the decap budget of SLP based method. This table shows the negative effect of violation time window drift to decap optimization.

C_{opt}	# Iter	V_{io}	# V_{io}	C	$\Delta C = (C - C_{\text{opt}})/C_{\text{opt}}$
82.4 pF	1	0.000507	26	80.3 pF	-2.5%
	2	0.0	0	113.1 pF	+38.82%

TABLE II
THE VIOLATION AND DECAP AMOUNT IN EACH ITERATIONS.

V. NEW METHOD BY PIECEWISE POLYNOMIAL MODELS

In this section, we present the new decap budgeting method by using more accurate piecewise polynomial method.

A. Voltage Waveform Prediction by Piecewise Polynomial

As we know that the violation residue is mainly caused by inaccuracy of L , we propose a more accurate piecewise polynomial based method to improve the prediction of voltage constraint L . The basic idea is to approximate the voltage waveform V_{opt} by piecewise polynomial instead of a simple straight line. However, this approximation is difficult because we don't know the curve V_{opt} until the optimal decap is added. Nevertheless, we still can observe some properties of curve V_{opt} , which will be helpful for our approximation :

- 1) $\frac{dV_{\text{opt}}}{dt}(t) \approx 0$ when $t \in [t_0, t_0 + \delta t]$; where δt is a small time;
- 2) $\frac{dV_{\text{opt}}}{dt}(t_1) \approx 0$;
- 3) $V_{\text{opt}}(t_1) \approx V_{\text{thre}}$.

Also we obtain another property $V_{\text{opt}}(t) \approx V(t_0)$ when $t \in [t_0, t_0 + \delta t]$ by integrating the equation in the first property.

After examining above properties, we find it's natural to approximate the curve V_{opt} by piecewise polynomial interpolation in two time regions : $[t_0, t_0 + \delta t]$ and $[t_0 + \delta t, t_1]$. In time region $[t_0, t_0 + \delta t]$, the voltage is approximated by the constant value because of property (1). In time region $[t_0 + \delta t, t_1]$, the voltage curve is approximated by a cubic polynomial determined by following interpolation conditions : $\frac{dV_{\text{opt}}}{dt}(t_0 + \delta t) = \frac{dV_{\text{opt}}}{dt}(t_1) = 0$, $V_{\text{opt}}(t_0 + \delta t) = V(t_0)$ and $V_{\text{opt}}(t_1) = V_{\text{thre}}$. So the approximating polynomial is defined as following:

$$\begin{aligned} P_0(t) &= V(t_0) \quad \text{if } t \in [t_0, t_0 + \delta t] \\ P_1(t) &= at^3 + bt^2 + ct + d \quad \text{if } t \in [t_0 + \delta t, t_1] \end{aligned} \quad (7)$$

where a, b, c, d are the interpolating coefficients.

Notice that variable δt remains unknown. Because this variable is critical to determine the shape of voltage curve, we estimate it through the voltage waveform we have already know. Suppose curve V_{k-1} is the waveform obtained in last optimizing iteration, as shown in Figure 5. We apply the same piecewise polynomial formula (7) to curve V_{k-1} but using different time window $[t_0, t_p]$, where $t_p = \text{argmin } V_{k-1}$. Similarly, a piecewise polynomial is obtained as following :

$$\begin{aligned} P'_0(t) &= V(t_0) \quad \text{if } t \in [t_0, t_0 + \delta t'] \\ P'_1(t) &= a't^3 + b't^2 + c't + d' \quad \text{if } t \in [t_0 + \delta t', t_p] \end{aligned}$$

In order to decide the variable $\delta t'$, we require the "distance" between voltage curve and piecewise polynomial approximation to be zero :

$$D = \int_{t_0}^{t_0 + \delta t'} (P'_0 - V_{k-1}) dt + \int_{t_0 + \delta t'}^{t_p} (P'_1 - V_{k-1}) dt = 0 \quad (8)$$

Solve equation (8) and obtain $\delta t'$:

$$\delta t' = \frac{2 \int_{t_0}^{t_p} V_{k-1} dt - (V(t_0) + V_{k-1}(t_p)) \cdot (t_p - t_0)}{V(t_0) - V_{k-1}(t_p)}$$

Then we assume $\delta t \approx \delta t'$ to generate the approximation of voltage waveform V_k by formula (7).


```

Determine the sampling nodes  $S$ 
Determine the violation time window  $[t_0, t_1]$ 
Build macro-model by equation (3)
Set intrinsic capacitance  $C_0 \leftarrow C_{intri}$ 
 $\mathbf{L}'_k \leftarrow \frac{1}{2}(\mathbf{V}(t_0) + V_{thre}) \cdot (t_1 - t_0)$ 
while  $True$  do
   $\mathbf{L} \leftarrow \mathbf{L}'_k$ 
   $\mathbf{V}(t_1) \leftarrow \mathbf{V}_p \leftarrow V_{thre}$ 
  while  $True$  do
    Obtain  $C_k$  by solving problem (4)
    Update  $\mathbf{V}_p$  by solving following equations :
      
$$(\mathbf{V}(t_0) - \mathbf{V}_p) \cdot C_k = A \cdot (\mathbf{V}(t_0) + \mathbf{V}_p)T + \mathbf{B}$$

      where  $T$  is  $(t_1 - t_0)/2$ 
      if  $|\mathbf{V}_p - \mathbf{V}(t_1)| \leq \epsilon$  then
         $Break$ 
      end
      if  $\mathbf{V}_p < V_{thre}$  then  $\mathbf{V}(t_1) = V_{thre}$ 
      else if  $\mathbf{V}_p \leq \mathbf{V}(t_1) - \delta$  then
         $\mathbf{V}(t_1) = \mathbf{V}(t_1) - \delta$ 
      else if  $\mathbf{V}_p \geq \mathbf{V}(t_1) + \delta$  then
         $\mathbf{V}(t_1) = \mathbf{V}(t_1) + \delta$ 
      else  $\mathbf{V}(t_1) = \mathbf{V}_p$ 
      end
     $C \leftarrow C_k + C_0$ 
    Get the voltage  $\mathbf{V}_{k-1}$  by solving equation
      
$$C \cdot \frac{d\mathbf{V}_{k-1}}{dt} = A \cdot \mathbf{V}_{k-1} + (G_{12}^T \cdot G_{11}^{-1} \cdot \mathbf{J}_1 - \mathbf{J}_2)$$

    Update  $\mathbf{L}'_k$  according to formula (9)
    if  $|\mathbf{L}'_k - \mathbf{L}| \leq \epsilon$  then
       $Break$ 
    end
  end
end

```

Algorithm 2: New Iterative SLP Based Algorithm

Finally, we can integrate piecewise polynomial approximation (7) to get voltage constraint $\mathbf{L}'_k$:

$$\begin{aligned}
\mathbf{L}'_k &\approx \int_{t_0}^{t_0+\delta t} \mathbf{P}_0 dt + \int_{t_0+\delta t}^{t_1} \mathbf{P}_1 dt \\
&= \frac{1}{2}(\mathbf{V}(t_0) + V_{thre}) \cdot (t_1 - t_0) + \frac{1}{2}(\mathbf{V}(t_0) - V_{thre})\delta t \\
&= \frac{1}{2}(\mathbf{V}(t_0) + V_{thre}) \cdot (t_1 - t_0) \\
&\quad + \frac{2 \int_{t_0}^{t_p} \mathbf{V}_{k-1} dt - (\mathbf{V}(t_0) + \mathbf{V}_{k-1}(t_p)) \cdot (t_p - t_0)}{2(\mathbf{V}(t_0) - \mathbf{V}_{k-1}(t_p)) / (\mathbf{V}(t_0) - V_{thre})}
\end{aligned} \tag{9}$$

Notice that $\mathbf{L}'_k$ becomes $\int_{t_0}^{t_1} \mathbf{V}_{opt} dt$ as voltage waveform $\mathbf{V}_{k-1} \rightarrow \mathbf{V}_{opt}$ because of $t_p \rightarrow t_1$ and $\mathbf{V}_{k-1}(t_p) \rightarrow V_{thre}$. So the voltage waveform $\mathbf{V}_{opt}$ is a “fix point” of equation (9) because using $\mathbf{L}' = \int_{t_0}^{t_1} \mathbf{V}_{opt}$ in SLP constraint (5) also leads to voltage waveform $\mathbf{V}_{opt}$. Therefore, the new voltage constraint $\mathbf{L}'_k$ equation (9) is convergent when $k \rightarrow \infty$.

B. The Flow of the New Decap Optimization Method

Plugging the new voltage constraint $\mathbf{L}'_k$ to linear programming problem (4), we could obtain a more accurate decap budget. After adding decap, the voltage waveforms are simulated to generate a better voltage constraint according to equation (9). The new iterative SLP-based decap optimization is shown in Algorithm 2.

We remark that the new iterative optimization flow is different from [12] in the following respects: 1. Sampling nodes, violation time window and macromodeling are determined once at the beginning, while algorithm 1 builds them in every iteration. 2. Violation time window is determined ignoring any capacitance to avoid violation time window drift while algorithm 1 considers decap already placed in the previous iterations, which causes violation time window drift. 3. In addition to voltages $\mathbf{V}(t_1)$, voltage constraints $\mathbf{L}$ are improved iteratively.

These differences are essential to eliminate the violation residue and avoid the violation time window drift.

VI. EXPERIMENTAL RESULTS

We implemented all three decap optimization algorithms in Matlab: sensitivity based decap optimization [4], SLP based algorithm 1 [12] and the new iterative SLP based algorithm 2 in Matlab. The built-in non-linear interior-point optimization function (“FMINCON”) is used in the sensitivity based algorithm. And another built-in linear programming function (“LINPROG”) is utilized in both SLP based algorithm 1 and the new iterative SLP algorithm. All the experiments were carried out on a Linux machine with 2.3GHz CPU and 8GB memory. However, due to the memory limit of 32-bit Matlab, only 2GB-3GB memory is available for our algorithm.

All test circuits are generated with realistic parameters for R, C and current sources based on industry designs. As mentioned in [12], the inefficiency of sensitivity based algorithm make it impossible to use very huge circuits. So we apply three algorithms to smaller test circuits to demonstrate the characteristic of each algorithms. We stress that the proposed method can easily improved by using partitioning scheme.

Table III shows the comparison between three algorithms, where the first column is the circuit name, the second column is the node number in the circuit, the third column is the violating node number. For each algorithm, iteration number, decap budget and runtime are also presented. Also the seventh column and eleventh column present the decap budget difference with respect to the decap values obtained by the proposed algorithm. All three algorithms remove all the voltage drop violation after adding decap. So the violating node number after adding decap is omitted.

As shown in the table, the proposed algorithm obtains almost the same (sometimes slightly less) decap values as the sensitivity based method. In contrast, the SLP based algorithm gives much larger decap values for most of cases.

We observe that all these decap budget over-estimations occur when the second iteration was carried out in the SLP based algorithm 1, which confirms our observation that over-estimation is caused by the violation time window drift in the second iteration. Usually, the violation time window drift after the first iteration is proportional to the decap added in the first iteration. Therefore, if violation is larger, the decap added in the first iteration will be larger and consequently lead to larger violation time window drift and larger decap overestimation.

		Sensitivity Based Algorithm					SLP Based Algorithm				Iterative SLP Algorithm		
Circuit	#Node	#Vio	#Iter	Budget	Δ	Time	#Iter	Budget	Δ	Time	#Iter	Budget	Time
ctk1	1049	77	31	82.4 pF	1.19%	158.8s	2	113.1 pF	38.82%	3.1s	2	81.4pF	3.1s
ctk2	2032	455	26	482.6 pF	-36.2%	127.1s	1	757.0 pF	0.00%	9.3s	1	757.0 pF	9.3s
ctk3	3532	1118	49	788.4 pF	-17.2%	302.5s	2	1305.7 pF	37.1%	23.1s	2	952.6 pF	43.2s
ctk4	4875	1413	51	1.043 nF	-10.0%	1020.8 s	2	2.089 nF	80.2%	39.4s	2	1.159 nF	81.6s
ctk5	6033	1733	18	1.198 nF	-0.80%	173.2 s	1	1.208 nF	0%	10.5s	1	1.208 nF	10.5s
ctk6	7550	2429	33	2.341 nF	-9.51%	904.7 s	2	4.720 nF	82.48%	26.4s	2	2.586 nF	45.5s
ctk7	8879	3250	44	2.965 nF	-11.62%	952.1 s	2	4.201 nF	25.2%	50.4s	2	3.355 nF	92.3s
ctk8	10251	4841	40	3.890 nF	4.62%	2165.9 s	2	5.836 nF	56.9%	113.8s	2	3.718 nF	208.7s
ctk9	20123	7098	NA	NA	NA	NA	1	8.134 nF	0%	65.5s	1	8.134 nF	47.9s
ctk10	40178	9741	NA	NA	NA	NA	2	15.973 nF	79.43%	64.3s	2	8.902 nF	97.4s

TABLE III
COMPARISON BETWEEN THE SENSITIVITY BASED ALGORITHM, THE SLP BASED ALGORITHM AND THE PROPOSED ALGORITHM

Vio	#Vio	Iter 1		Iter 2 in SLP Based Algorithm		Iter 2 in Iterative SLP Algorithm		overestimation
		Vio	#Vio	Budget	Vio	#Vio	Budget	
0.0041	77	0.000507	26	80.3 pF	0.0	0	113.1 pF	38.9%
0.0160	203	0.0015	55	170.2 pF	0.0	0	241.8 pF	34.3%
0.0502	386	0.0039	104	226.2 pF	0.0	0	375.3 pF	52.3%
0.1183	546	0.0066	138	284.4 pF	0.0	0	556.1 pF	73.3%

TABLE IV
VIOLATION AND DECAP BUDGET IN EACH ITERATIONS

Table IV shows the violation and the decap in each iterations of both the SLP based algorithm and the proposed algorithm. The test circuit “ctk1” is used in this experiment and the current source is scaled up to increase the violation. The column “Vio” is the total violation, i.e. $Vio = \sum_{i \in vio} \int V dt$, where t in violation time window. And “#Vio” is the violating node number. Although the violation in practice is not as large as shown in table IV, table IV shows the larger the violation is, the larger the decap overestimation is.

Although the scale of these test circuits are limited due to the memory limit of 32-bit MATLAB, The table still shows a good comparison of CPU runtime between the proposed and the SLP based algorithm, both are significantly less than the sensitivity based algorithm. We also can see the the proposed algorithm is sometimes slower than the SLP based algorithm despite they both complete the optimization in 2 iterations. This is because the LP problem in the 2nd iteration remains as large as in the first iteration for our algorithm but the 2nd iteration of the SLP based algorithm has a reduced LP problem to solve.

VII. CONCLUSION

In this paper, we have proposed an efficient decoupling (decaps) capacitance optimization algorithm. The new method improves the existing method [12], by applying the more accurate piecewise polynomial models to estimate the voltage noises during the linear programming processes. The proposed method has the efficiency of the charge-based methods and the accuracy of the sensitivity-based methods. Experimental results have demonstrated that the proposed method leads to the much smaller decap values, which are similar to the best-known sensitivity-based method, than the existing charge based method, and while it maintains the same efficiency of the charge-based method.

REFERENCES

- [1] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*. Addison-Wesley Publishing Company, 1990.
- [2] Y. Cai, J. Fu, X. Hong, S. X.-D. Tan, and Y. Luo, “Power/ground network optimization considering decap leakage currents,” *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 53, no. 10, pp. 1012–1016, Oct. 2006.
- [3] H. H. Chen and D. D. Ling, “Power supply noise analysis methodology for deep-submicron VLSI chip design,” in *Proc. Design Automation Conf. (DAC)*, 1997.
- [4] J. Fu, Z. Luo, X. Hong, Y. Cai, S. X.-D. Tan, and Z. Pan, “A fast decoupling capacitor budgeting algorithm for robust on-chip power delivery,” *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Science(IEICE)*, vol. E87-A, no. 12, pp. 3273–3280, Dec. 2004.
- [5] L. Kang, Y. Cai, Y. Zou, J. Shi, X. Hong, and S. X.-D. Tan, “Fast decoupling capacitor budgeting for power/ground network using random walk approach,” *ASP-DAC '07: Proceedings of the 2007 conference on Asia South Pacific design automation*, Jan 2007.
- [6] H. Li, J. Fan, Z. Qi, S. X.-D. Tan, L. Wu, Y. Cai, and X. Hong, “Partitioning-based approach to fast on-chip decoupling capacitor budgeting and minimization,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 11, pp. 2402–2412, Nov. 2006.
- [7] S. Pant and E. Chiprout, “Power grid physics and implications for cad,” in *Proceedings of 2006 Design Automation Conference*, 2006, pp. 199–204.
- [8] H. Su, S. S. Sapatnekar, and S. R. Nassif, “Optimal decoupling capacitor sizing and placement for standard cell layout designs,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, pp. 428–436, Apr 2003.
- [9] H. Su, S. Sapatnekar, and S. Nassif, “An algorithm for optimal decoupling capacitor sizing and placement for standard cell layouts,” *ISPD '02: Proceedings of the 2002 international symposium on Physical design*, Apr 2002.
- [10] K. Wang and M. Marek-Sadowska, “On-chip power supply network optimization using multigrid-based technique,” in *Proc. Design Automation Conf. (DAC)*, 2003, pp. 113–118.
- [11] C. K. S. Zhao and K. Roy, “Decoupling capacitance allocation and its application to power-supply noise-aware floorplanning,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, pp. 81–92, Jan 2002.
- [12] M. Zhao, R. Panda, S. Sundareswaran, S. Yan, and Y. Fu, “A fast on-chip decoupling capacitance budgeting algorithm using macromodeling and linear programming,” *Design Automation Conference, 2006 43rd ACM/IEEE*, pp. 217 – 222, Jun 2006.