

Potential Slack: An Effective Metric of Combinational Circuit Performance

Chunhong Chen, Xiaojian Yang and Majid Sarrafzadeh

Department of Electrical and Computer Engineering
Northwestern University, Evanston, IL 60208-3118

Abstract

This paper proposes the concept of potential slack and show it is an effective metric of combinational circuit performance. We provide several methods for estimating potential slack and prove one (a maximal-independent-set based algorithm) in particular works best. Experiments in gate sizing show that potential slack provides 100% correct prediction for circuit area optimization. We also explore the role of potential slack in timing-driven placement.

1 Introduction

For a long time, circuit delay, area cost and power dissipation have been major optimization objectives in designing integrated circuit chips. In general, faster circuits suffer from the higher area/power penalty. One interesting question is that for a specific circuit, how much additional area and/or power is required per unit delay improvement, or to what extent the timing performance degrades when the area and/or power is reduced by one unit. The answer is strongly related to non-critical paths of a circuit, since the delay penalty caused by area/power improvement of logic modules on non-critical paths will not necessarily increase circuit delay.

From a timing standpoint, non-critical paths can be characterized by timing *slack* of their constituent logic modules. Experience shows that circuit implementations with same delay, area and/or power can have totally different module slacks. Also, traditional optimization approaches do not directly deal with slack. This motivates us to take a closer look at slack and to devise an effective way of measuring it. From the typical gate-level optimization tradeoff curve of delay and area/power, the logic module with smaller area/power has longer delay. The slack for a module provides an upper bound of its delay increase without violating timing constraints and, hence, represents a “*potential*” capability of obtaining area/power reduction. On the other hand, at physical level, slack can be interpreted as a measure of wiring “freedom” (opposed to delay “constraints”) in timing-driven placement and routing, and hence as a predictor of final circuit timing.

A well-known technique for slack management is the *Zero-Slack Algorithm* (ZSA) [1]. Its improved versions of for applications in placement have also been proposed, e.g., in [2, 3]. As its name implies, *potential slack* is the slack that can potentially be used for optimizing a circuit (the exact definition will be given in the next section). In this paper, we claim that potential slack is a very effective metric of combinational circuit performance. We explore

several ways of estimating potential slack of a circuit and present an *optimal, polynomial-time* algorithm. Applications to gate sizing and placement are provided to show that potential slack serves as a good predictor of circuits’ performance.

2 Preliminaries

Consider a combinational circuit which consists of a set of *modules* $\mathbf{M} = \{m_1, m_2, \dots, m_n\}$, and a set of *nets* $\mathbf{N} = \{n_1, n_2, \dots, n_l\}$. Naturally, we associate the circuit with a *directed-acyclic graph* $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, where a node $v \in \mathbf{V}$ denotes a module, and there is an edge $e \in \mathbf{E}$ from node v_i to v_j if v_j is an immediate *fanout* of v_i . Also, each node v is associated with a *delay* $d(v)$, which stands for the delay of node v itself plus the interconnection delay. A *vector* of delays $\mathbf{D}(\mathbf{V}) = [d(v_1) d(v_2) \dots d(v_n)]$ is called *delay distribution* of the circuit. For convenience, suppose that the *arrival times* for all primary inputs are zero and that the *required times* for all primary outputs are the given timing constraints. A well-known procedure to compute arrival time, $a(v)$, and required time, $r(v)$, for each node v is given recursively by

$$\left. \begin{aligned} a(v) &= \max_{u \in FI(v)} (a(u) + d(v)) \\ r(v) &= \min_{w \in FO(v)} (r(w) - d(w)) \end{aligned} \right\} \quad (1)$$

where $FI(v)$ is a set of fanins of node v , and $FO(v)$ a set of fanouts. *Slack* of node v is defined as $s(v) = r(v) - a(v)$. A vector of slacks $\mathbf{S}(\mathbf{V}) = [s(v_1) s(v_2) \dots s(v_n)]$ is called *slack*

distribution of the circuit, and $|\mathbf{S}(\mathbf{V})| = \sum_{i=1}^n s(v_i)$ is called

total slack. The circuit is said to be *safe* if and only if $\mathbf{S}(\mathbf{V}) \geq \mathbf{0}$. A *slack assignment* is a vector of *incremental* delays $\Delta\mathbf{D}(\mathbf{V}) = [\Delta d(v_1) \Delta d(v_2) \dots \Delta d(v_n)] \geq \mathbf{0}$ which updates the delay distribution from $\mathbf{D}(\mathbf{V})$ to $\mathbf{D}_\Delta(\mathbf{V}) = \mathbf{D}(\mathbf{V}) + \Delta\mathbf{D}(\mathbf{V})$, and hence updates the slack distribution from $\mathbf{S}(\mathbf{V})$ to $\mathbf{S}_\Delta(\mathbf{V})$. If $\mathbf{S}_\Delta(\mathbf{V}) \geq \mathbf{0}$, then the slack assignment $\Delta\mathbf{D}(\mathbf{V})$ is said to be

effective, and $|\Delta\mathbf{D}(\mathbf{V})| = \sum_{i=1}^n \Delta d(v_i)$ is thereby called

effective slack of the assignment. The maximum effective slack for all possible slack assignments is called *potential slack* of the circuit. We use $\Delta_n\mathbf{D}(\mathbf{V})$ to represent the slack assignment which results in potential slack. For any assignment, if there exists a slack $s(v_i) > 0$ ($1 \leq i \leq n$) in its resultant slack distribution, one is always able to increase effective slack by assigning to node v_i an additional delay

$\Delta d_i = s(v_i)$ such that $s(v_i) = 0$ after the assignment. In other words, if a slack assignment $\Delta_m \mathbf{D}(\mathbf{V})$ leads to potential slack denoted by PS , i.e., $PS = |\Delta_m \mathbf{D}(\mathbf{V})| = \max_{\Delta} |\Delta \mathbf{D}(\mathbf{V})|$ subject

to $S_{\Delta}(\mathbf{V}) \geq \mathbf{0}$, then the resultant slack distribution $S_{\Delta m}(\mathbf{V}) = \mathbf{0}$.

Fig. 1 shows a four-node example where the delay distribution is assumed to be $\mathbf{D}(\mathbf{V}) = [1 \ 1 \ 1 \ 1]$. Initially, the slack distribution is $\mathbf{S}(\mathbf{V}) = [5 \ 5 \ 5 \ 5]$, and the total slack is $|\mathbf{S}(\mathbf{V})| = 20$. $\Delta_1 \mathbf{D}(\mathbf{V}) = [2 \ 1 \ 1 \ 1]$ is an effective slack assignment since $S_{\Delta_1}(\mathbf{V}) = [1 \ 2 \ 1 \ 1] > \mathbf{0}$. $\Delta_m \mathbf{D}(\mathbf{V}) = [5 \ 5 \ 0 \ 0]$ leads to potential slack $PS = |\Delta_m \mathbf{D}(\mathbf{V})| = 10$, and $S_{\Delta m}(\mathbf{V}) = \mathbf{0}$.

Since $s(v_i)$ is the upper bound of incremental delay for node v_i while keeping safety of \mathbf{G} , any effective slack is no more than total slack of \mathbf{G} . In Fig. 1, for example, $PS = 10$, while total slack $|\mathbf{S}(\mathbf{V})| = 20$. For a given circuit, while calculation of total slack is straightforward, finding PS is a nontrivial task. As a timing characterization of a circuit, PS represents a *real* total delay budget one can get from the circuit. A slack assignment which generates PS is called *optimal slack assignment*.

3 Potential Slack Estimation and Applications

In this section we first briefly review the well-known zero-slack algorithm (ZSA), and then describe a new maximal-independent-set based algorithm (MISA) with applications.

3.1 ZSA

Given a graph, ZSA starts with nodes of minimum positive slack and locally performs slack assignment such that their slacks become zero [1]. More specifically, at each iteration, ZSA identifies a path on which all nodes have minimum slack (denoted by s_{min}), then assigns each node an additional delay s_{min}/N_{min} , where N_{min} is the number of nodes on the path. In Fig. 1, for example, path $\{v_1, v_3, v_4\}$ is first identified, and each node on the path is assigned an additional delay $5/3$. Slacks of all nodes in the figure except v_2 become zero, while slack of v_2 is updated as $5/3$. After assigning additional delay of $5/3$ again to v_2 , the algorithm terminates with effective slack of $20/3$ (note that maximum effective slack of Fig. 1 is 10). This example shows that while ZSA is simple and easy to implement, it is far from optimal slack assignment.

3.2 MISA

ZSA generates effective slack assignment in a greedy way, ignoring the possible effect of a node delay increase on the slack of its transitive fanins/fanouts. To take this effect into account, let us analyze the slack relation by looking at two cases below: (i) node v with a fanin node v_{in} . If an additional delay $\Delta d(v)$ is assigned to v , the slack of v is decreased by $\Delta d(v)$. Meanwhile, the slack of v_{in} will be affected (reduced) only if

$$r(v) - r(v_{in}) - d(v) < \Delta d(v) \quad (2)$$

(ii) node v with a fanout node v_{out} . If we assign an additional delay $\Delta d(v)$ to it, the slack of v is decreased by $\Delta d(v)$. However, the slack of v_{out} will be reduced only if

$$a(v_{out}) - a(v) - d(v_{out}) < \Delta d(v) \quad (3)$$

Node v_{in} (or v_{out}) is said to be *slack-sensitive* to node v if

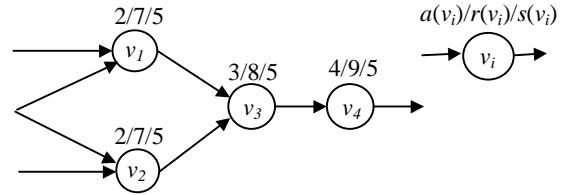


Figure 1. Slack distribution and slack assignment

inequality (2) (or (3)) holds true. In particular, if $r(v) - r(v_{in}) = d(v)$ (or $a(v_{out}) - a(v) = d(v_{out})$), then fanin node v_{in} (or fanout node v_{out}) of node v is slack-sensitive to v for any $\Delta d(v) > 0$. Here $\Delta d(v)$ can be viewed as a delay *benefit* which contributes to the effective slack of the circuit. In contrast, the slack reduction of node v 's fanins/fanouts represents a slack *penalty*, which prevents further benefit obtained from them. Intuitively, a good slack assignment should maximize the benefit and minimize the penalty.

In order to reduce slack penalty, it is desirable to choose a node v with larger $r(v)$ and/or smaller $a(v)$ such that the number of its fanins and fanouts which are slack-sensitive to v can be minimized (see (2) and (3)). This leads us to first select nodes with maximum slack (opposite to minimum slack used in ZSA) for slack assignment in a given circuit. For those nodes with maximum slack (denoted by s_m) in $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, we construct a *slack-equalization graph* $\mathbf{G}_m = (\mathbf{V}_m, \mathbf{E}_m)$, where $\mathbf{V}_m = \{v \mid v \in \mathbf{V} \text{ and } s(v) = s_m\}$ and $\mathbf{E}_m = \{(u, v) \mid (u, v) \in \mathbf{E}, \text{ and } u, v \text{ are slack-sensitive}\}$. A *transitive slack-equalization graph* $\mathbf{G}_t = (\mathbf{V}_m, \mathbf{E}'_m)$ of \mathbf{G}_m is a directed graph such that there is an edge $(u, v) \in \mathbf{E}'_m$ if and only if there is a directed path from u to v in \mathbf{G}_m . Thus, a *maximal independent set* (MIS) of \mathbf{G}_t corresponds to a set of nodes $\mathbf{V}_{MIS} \subseteq \mathbf{V}_m$ such that the number of nodes which are slack-sensitive to any node $v \in \mathbf{V}_{MIS}$ is minimized. In an extreme case where $\Delta d(v) = s_m$ (note that for the timing constraints to be met, $\Delta d(v)$ is no more than s_m), all transitive fanins/fanouts of node v are slack-sensitive to v . Let s_{m-1} be the second largest slack in \mathbf{G} . If we choose $\Delta d(v) = s_m - s_{m-1}$, then, from (2) and (3), no nodes with slack less than s_m are slack-sensitive to v , and the slack penalty is thus minimized. We describe a MIS-based algorithm for slack assignment below.

MIS-based Slack Assignment Algorithm (MISA) {

Input: graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ and given timing constraints

Output: slack assignment $\Delta \mathbf{D}(\mathbf{V})$

Begin

Compute slack for each node $v \in \mathbf{V}$ and initialize $\Delta \mathbf{D}(\mathbf{V}) = \mathbf{0}$;

Find s_m and s_{m-1} in \mathbf{G} and let $\delta = s_m - s_{m-1}$;

While ($\delta > 0$) {

Construct a transitive slack-equalization graph \mathbf{G}_t ;

Find maximal independent set **MIS** of \mathbf{G}_t ;

Assign an additional delay δ to each node in **MIS**, i.e.,

$$\Delta d(u) \leftarrow \Delta d(u) + \delta, \quad \forall u \in \mathbf{MIS} ;$$

Update the node slack and δ in \mathbf{G} ;

}

End

}

Fig. 2 shows an 8-node example graph, where the initial delay distribution is assumed to be one unit, i.e., $\mathbf{D}(\mathbf{V}) = \mathbf{1}$. By inspection, ZSA generates slack assignment $\Delta_{\text{ZSA}}\mathbf{D}(\mathbf{V}) = [3.5 \ 1.5 \ 1.5 \ 1.5 \ 1.5 \ 2 \ 2 \ 0]$ which leads to effective slack $|\Delta_{\text{ZSA}}\mathbf{D}(\mathbf{V})| = 13.5$. By applying MISA instead, we can obtain an effective slack assignment $\Delta_{\text{MISA}}\mathbf{D}(\mathbf{V}) = [2 \ 0 \ 3 \ 0 \ 3 \ 5 \ 5 \ 0]$ with $|\Delta_{\text{MISA}}\mathbf{D}(\mathbf{V})| = 18$ which is 33% improvement compared to $|\Delta_{\text{ZSA}}\mathbf{D}(\mathbf{V})|$.

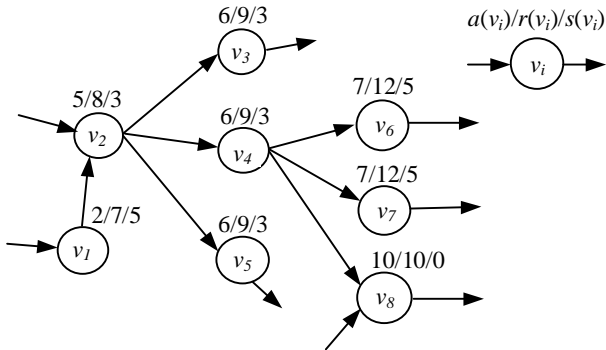


Figure 2. Slack calculation

To look at, in general, the difference between effective slacks resulting from ZSA and MISA, let us consider a node v with f fanouts. Suppose that all nodes have same slack s . While ZSA generates effective slack $|\Delta_{\text{ZSA}}\mathbf{D}(\mathbf{V})| = s + (f - 1)s/2$, MISA results in effective slack $|\Delta_{\text{MISA}}\mathbf{D}(\mathbf{V})| = f \cdot s$. Hence, $(|\Delta_{\text{MISA}}\mathbf{D}(\mathbf{V})| - |\Delta_{\text{ZSA}}\mathbf{D}(\mathbf{V})|) / |\Delta_{\text{ZSA}}\mathbf{D}(\mathbf{V})| = (f - 1)/(f + 1)$, meaning $|\Delta_{\text{MISA}}\mathbf{D}(\mathbf{V})|$ is almost twice as $|\Delta_{\text{ZSA}}\mathbf{D}(\mathbf{V})|$ when f is large. This shows that MISA can provide significant improvement over ZSA, in terms of effective slack, for large number of fanouts.

Based on the above discussions, we have following theorems without proof.

Theorem 1: *If an optimal slack assignment is defined to be a slack assignment which results in potential slack (i.e. maximum effective slack), then MISA produces an optimal slack assignment.*

Theorem 2: *The time complexity of MISA is $O(K \cdot n^3)$, where n and K are the number of nodes and the number of different slacks in a given graph, respectively.*

3.3 Application to Gate Sizing

Potential slack can be used for a class of delay-constrained area/power optimization problems [5-7, 12, 13]. Given the timing constraints which represent circuit performance requirement, *gate sizing* problem is to find a set of gates/nodes such that their physical sizes can be reduced (by using smaller *cell instances* from a *target library*) for area/power minimization without violating timing constraints. Since a gate with smaller size has longer delay and hence requires more slack, area/power reduction by gate sizing is strongly determined by slack assignment of the corresponding graph. For specific circuits, high potential slack promises significant area/power reduction during gate sizing (note that this reduction may not be maximized due to the discrete nature of cell sizes available in the library).

3.4 Application to Placement

Another application of potential slack is timing-driven placement design [8]. Given the timing constraints of a gate-level circuit implementation, potential slack can give us a basic idea about the freedom/flexibility of all the signal nets of the circuit during placement. Intuitively, high potential slack of a circuit allows a large number of signal nets to be routed easily without violating the timing performance. We first generate several implementations for a specific circuit and then perform timing-driven placement to obtain more accurate circuit delay.

4 Experiments

We have implemented both ZSA and MISA on top of SIS package [4]. We tested each algorithm on different gate-level implementations of benchmark circuits by using five different mappers (with a standard cell library). Here we selected three commands from SIS for technology mapping: *map -m 1.0*, *map -AF -m 1.0*, and *map -m 0.0*. The three implementations are denoted by I_1 , I_2 and I_3 , respectively.

We performed a first set of experiments to show the longest path delay (LPD), total slack and potential slack (PS) for different implementations and their comparison. The results are summarized in Table 1. For all circuits, the potential slack is much less than total slack, and larger total slack (or smaller LPD) does not always result in larger PS. For instance, circuits *C432* and *dalu* have their minimum LPD and maximum total slack under I_2 , while their PSs are highest under I_1 .

Experimental results of MISA followed by gate sizing are presented in Table 2. From Tables 1 and 2, we see that for all tested circuits, the implementation with maximum PS leads to maximum area reduction. In other words, potential slack provides 100% “correct” prediction. In contrast, LPD and total slack result in 20% and 40% (on average) chance of obtaining correct prediction, respectively (see Tables 1 and 2 for details). This is important since we can predict and/or maximize the possible area/power reduction by dealing with potential slack without having to go through lower level optimization. Typically, potential slack estimation is dozens of times faster than gate sizing in our experiments. Experiments with potential slacks estimated by ZSA and MISA show that MISA produces about 30% (on average) more effective slack than ZSA.

Also, we carried out placement experiment on the three implementations of each tested circuit by using the timing-driven placement tool from *TimberWolf* [8] (we took its latest commercial version — *Itools* 1.4.0 [11]). Given the uniform pin-pair (i.e., between each primary input and each primary output) timing constraints T_{spec} for each circuit, the tool calculates the *timing penalty* $P(p) = T_d(p) - T_{spec}$ for each path p , where T_d is the path delay. The total timing penalty P_T is the sum of $P(p)$ over all critical paths. Results show that for most circuits (except several small circuits), higher PS corresponds to smaller total timing penalty which translates into better timing performance of the post-layout circuit. For small circuits, however, the total timing penalty is dominated by LPD unless there is significant difference

circuit	size (# gates)*	I_1 (implementation #1)			I_2 (implementation #2)			I_3 (implementation #3)		
		LPD	total slack	PS	LPD	total slack	PS	LPD	total slack	PS
<i>tcon</i>	25	7.8	348.4	228.6	7.0	373.6	174.4	10.8	187.2	132.7
<i>decod</i>	34	8.2	89.4	21.7	8.4	73.0	20.8	9.0	76.6	20.5
<i>z4ml</i>	41	19.0	176.6	99.5	19.2	175.0	82.2	17.4	170.2	69.4
<i>cmb</i>	45	12.6	190.2	87.2	12.4	167.2	78.8	11.6	200.0	64.8
<i>sct</i>	81	46.0	1903.8	745.4	43.8	2240.4	734.4	42.4	1703.6	468.4
<i>9symml</i>	199	23.0	2190.2	730.6	24.2	2398.8	810.3	28.8	895.8	387.2
<i>C432</i>	309	77.2	3259.6	2061.6	73.4	3645.8	1837.0	74.4	3159.0	1686.6
<i>i9</i>	422	62.2	31683.6	8812.9	33.0	41807.0	11559.0	91.2	16381.8	6408.0
<i>C1908</i>	439	54.0	6514.0	1692.9	53.4	6640.0	1637.9	55.8	4159.6	1155.8
<i>dalu</i>	1461	114.0	103066.8	29943.0	77.2	135144.2	28354.3	134.4	46670.4	18795.4

* This is the number of gates in the circuit under I_1 .

Table 1. Comparison of different implementations on a set of benchmark circuits

circuit	area reduction (%)		
	I_1	I_2	I_3
<i>tcon</i>	31.2	14.8	0.0
<i>decod</i>	26.4	22.7	2.5
<i>z4ml</i>	20.4	14.3	2.7
<i>cmb</i>	13.7	10.3	1.5
<i>sct</i>	21.6	14.8	5.2
<i>9symml</i>	15.9	16.7	1.0
<i>C432</i>	16.4	9.3	2.7
<i>i9</i>	18.7	24.0	5.5
<i>C1908</i>	22.2	16.2	2.3
<i>dalu</i>	19.6	15.3	2.7
average	20.1	15.8	2.6

Table 2. Application of potential slack to gate sizing

between the PSs of two implementations. The reason is that typically, wire delay in these circuits only accounts for a small portion of critical path delay (note that our placement data are omitted here due to space limitation).

5 Conclusions

We have introduced the notion of potential slack for measuring potential area/power reduction and flexibility of physical design, and proposed a maximal-independent-set based algorithm for estimating the potential slack of given circuits. It has been shown that potential slack provides another metric which measures circuit performance that traditional metrics would not be able to do. In terms of area optimization in gate sizing application, potential slack provides 100% correct prediction, compared to 20%-40% chance of obtaining correct prediction by other metrics. In terms of timing performance in placement application, potential slack can be used as an important metric. Further work is under way, including applications to other optimization problems, fast estimation of potential slack, and logic synthesis for maximum potential slack.

References

- [1] R. Nair, C. L. Berman, P. S. Hauge, and E. J. Yoffa, "Generation of Performance Constraints for Layout," *IEEE Transactions on Computer-Aided Design*, CAD-8(8): 860-874, August 1989.
- [2] T. Gao, P. M. Vaidya, and C. L. Liu, "A New Performance Driven Placement Algorithm," in *Proceedings of ICCAD*, pp.44-47, 1991.
- [3] H. Youssef and E. Shragowitz, "Timing Constraints for Correct Performance," in *Proceedings of ICCAD*, pp.24-27, 1990.
- [4] E. M. Sentovich *et al.*, "SIS: A System for Sequential Circuit Synthesis," *Technical Report UCB/ERL M92/41*, Univ. of California, Berkeley, May 1992.
- [5] C. Chen and M. Sarrafzadeh, "An Effective Algorithm for Gate-Level Power-Delay Tradeoff Using Two Voltages," in *Proceedings of ICCD*, pp.222-227, 1999.
- [6] P. Girard, C. Landraut, S. Pravossoudovitch, and D. Severac, "A Gate Resizing Technique for High Reduction in Power Consumption," in *Proceedings of International Symposium on Low Power Electronics and Design*, pp.281-286, 1997.
- [7] H. R. Lin and T. Hwang, "Power Reduction by Gate Sizing with Path-Oriented Slack Calculation," in *Proceedings of IEEE ASP-DAC'95/CHDL'95/VLSI'95*, pp.7-12, June 1995.
- [8] W. Swartz and C. Sechen, "Timing Driven Placement for Large Standard Cell Circuits," in *Proceedings of DAC*, 1995.
- [9] J. Cong, L. He, K. Khoo, C. Koh and Z. Pan, "Interconnection Design for Deep Submicron ICs," in *Proceedings of ICCAD*, pp.478-485, 1997.
- [10] R. H. Moehring, "Graphs and Orders: the Role of Graphs in the Theory of Ordered Sets and Its Applications," published by D. Reidel Publishing Company, edited by I. Rival, New York and London, pp.41-101, May 1984.
- [11] *Itools* 1.4.0 (formerly *TimberWolf*): <http://www.internetcad.com>.
- [12] C. Chen and M. Sarrafzadeh, "Provably Good Algorithm for Low Power Consumption with Dual Supply Voltages," in *Proceedings of ICCAD*, pp.76-79, 1999.
- [13] C. Chen and M. Sarrafzadeh, "Power Reduction by Simultaneous Voltage Scaling and Gate Sizing," in *Proceedings of ASPDAC*, pp.333-338, 2000.

