

# Computation of Yield Gradients from Statistical Timing Analysis

Vladimir Zolotov  
IBM Watson Research Center  
Yorktown Heights, NY  
zolotov@us.ibm.com

Jinjun Xiong<sup>\*</sup>  
University of California  
Los Angeles, CA  
jinjun@ucla.edu

Chandu Visweswariah  
IBM Watson Research Center  
Yorktown Heights, NY  
chandu@us.ibm.com

## ABSTRACT

Statistical timing is an efficient way of taking into account process variations during performance analysis. However, optimizing a circuit across the entire process space is an extremely difficult challenge. Traditionally, static timing has been useful both for timing sign-off and to provide diagnostics for optimization. Traditional diagnostics such as the notion of a unique critical path or timing slack can no longer be used as metrics to guide optimization in the presence of variability. This paper presents a novel and efficient method to compute the gradient of parametric yield with respect to the delay of each gate or wire. The resulting gradients can be rank-ordered for discrete optimization in a physical synthesis setting, or fed to a nonlinear optimizer for continuous optimization of design parameters such as transistor sizes, thus enabling formal mathematical yield optimization.

## 1. INTRODUCTION

Timing optimization is traditionally conducted by physical synthesis in a library-based flow or transistor sizing in a transistor-level custom design methodology. The optimization in either case is guided by deterministic timing slack or margins. In the presence of process variations, the goal is to close timing in the entire process space. Traditional metrics like deterministic slack fail to provide correct guidance to optimizers.

To help optimization, a concept called *criticality* was proposed in [7], but not computed correctly due to certain correlations being ignored. A related concept called *criticality index* was suggested in the context of PERT networks [5]. In [6], a method was proposed to compute the gradient of the mean of circuit delay with respect to the mean delay of each edge of the timing graph, which was proved to be numerically the same as criticality. However, this work did not compute the sensitivity of the variance of circuit delay. If such gradients are used to optimize a circuit, optimal mean circuit delay can be obtained, but variance and actual yield will be unpredictable.

In this paper, we propose a novel and efficient method of computing yield gradients as a post-processing step after statistical timing has been completed. We compute the gradient of *parametric timing yield* (including consideration of variance) with respect to each timing edge. The yield gradients (in conjunction with statistical slacks or timing mar-

gins) directly enable various types of optimization such as parametric yield maximization or power minimization subject to yield and timing requirements.

In our approach we use the same concept of graph cutsets as [2], but we do not use any perturbation of timing edge PDFs or convolution. Instead, we derive explicit analytical expressions for all components of yield gradients, which makes our approach efficient. Unlike [6] we do not use any chain-ruling or propagation of sensitivities through the timing graph, which in turn greatly simplifies our computations.

The organization of the rest of the paper is as follows. Section 2 points out the limitations of both traditional deterministic slack and statistical slack as metrics to guide optimization. Section 3 reviews a criticality computation procedure described in a companion paper, upon which the proposed method is built. The heart of the derivation of yield gradients can be found in Section 4, making use of formulas derived in Appendix A. Potential applications of these yield gradients are described in Section 5.

This is a concept paper representing work in progress, and hence does not contain numerical results. However, the authors believe this work to be valuable in providing insight and setting research directions.

## 2. SLACK IS A POOR METRIC

Timing closure in the presence of process variations is a nightmare. Traditional incremental timing is run at a chosen process corner to guide the optimization, but timing sign-off requires either multi-corner timing or statistical timing. If sign-off timing is not achieved, the optimization continues with some annotations from the sign-off timing as to which cones of logic need improvement. The trouble is that as soon as timing is closed at process corner  $A$ , violations are seen at process corner  $B$ , and vice versa, and the target remains elusive.

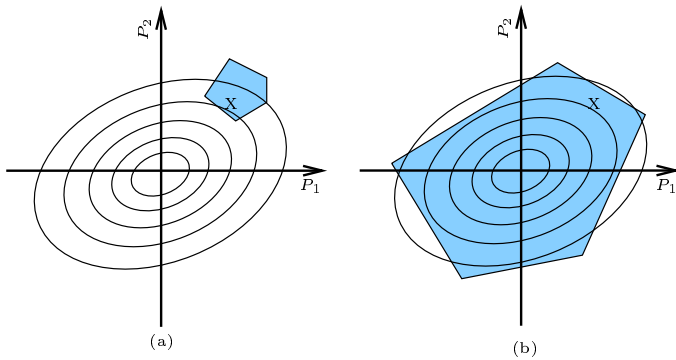
Traditional timing provides two important diagnostics:

1. The identity of the critical path, which is the location of the most “bang for the buck” during optimization.
2. Timing slack or margin.

Unfortunately, in the presence of process variations, neither of these diagnostics is useful. Each point in the process space can have a unique critical path, so in reality there is a set of critical paths, each of which has a non-zero probability of being critical. For illustration purposes, Fig. 1 shows a 2-dimensional space with process parameters  $P_1$  and  $P_2$ , along with contours of equal probability. Realistic situations

---

<sup>\*</sup>This work was done while this author was an intern at IBM Research.



**Figure 1: Criticality of a critical path in (a) a small region of the process space; (b) a large region of the process space.**

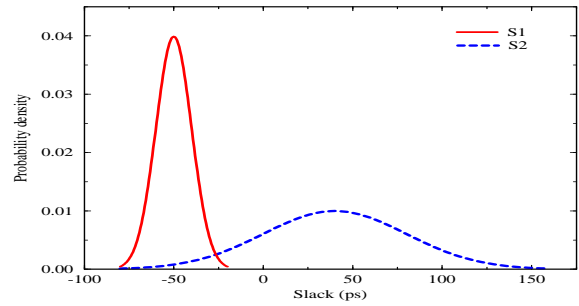
will have a much higher dimensionality. The  $X$  in the figure shows the process corner at which timing is conducted in order to guide optimization, at which a particular path is found to be the most critical. As shown in Fig. 1a, this path may be critical in only a small portion of the process space around  $X$ , or it may be critical in much of the high-probability process space as shown in Fig. 1b. Thus it is unclear from deterministic timing whether or not it is important to improve the timing characteristics of this path.

One quickly comes to the conclusion that one requires either multi-corner slack or statistical slack to guide optimization. Multi-corner slacks cannot be obtained incrementally and an exponential number of corners in the process space makes this procedure prohibitive. Instead, statistical slack is more conducive to incremental operation of the timer, as described in [7]. However, statistical slack also has problems in serving as a good metric for optimization.

For one thing, statistical slack is a distribution, and present-day optimization tools typically do not know how to deal with distributions. For another, statistical slack is richer in information content than a distribution since it is typically parameterized by the sources of variation (e.g., a first-order canonical form [7]). On the one hand, this opens up possibilities of using this parameterized model to *choose* the type of optimization that would best improve timing characteristics, but, on the other, it makes it even more complicated for the optimizer to use statistical slack as a metric.

One possibility is to use a  $\sigma$ -sampled value of the slack as a metric, e.g., the  $\mu - 3\sigma$  value of the slack. This has the advantage of ensuring adequate parametric yield when timing is closed, and being a single number that represents the entire process space. Unfortunately, this metric has problems, too. Fig. 2 shows two slack distributions  $S1$  and  $S2$  which have the same  $-3\sigma$  value. In the case of  $S1$ , the distribution is pretty tight and improvement (i.e., movement to the right) is best accomplished by moving the entire distribution. Moving an entire distribution typically costs area and power since it is often achieved by inserting buffers or up-sizing gates. On the other hand,  $S2$  can be improved by reducing its sensitivity to process, i.e., tightening its distribution. Unfortunately,  $\sigma$ -sampled slack does not give us this type of insight.

Now we will consider one last example to demonstrate the problems with statistical slack. Consider a situation in



**Figure 2: Statistical slack of two paths with the same  $-3\sigma$  value.**

which two  $\sigma$ -sampled path slacks are  $-70$  and  $-50$  ps, respectively. Traditional wisdom says to improve the first path till its slack reaches  $-50$  ps, and then try to improve both. Depending on the spread and correlation of the two distributions, this may not be so wise. If the spread of the slacks is large and the two slacks are uncorrelated, then the second path may be almost as much of a yield limiter as the first. If they are tightly correlated, however, then traditional wisdom is sound.

Some of the desired features of a good diagnostic are:

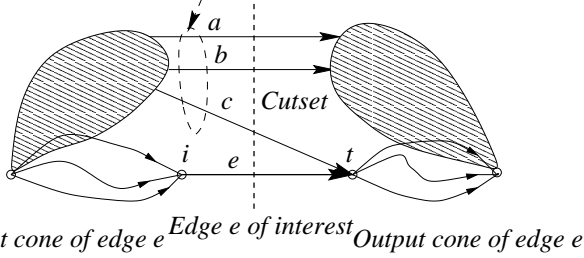
- preferably, a single number;
- preferably, a number with a fixed range such as 0 to 1;
- a number that implicitly represents criticality in the entire process space;
- a number that is implicitly correlation-aware; and
- a number that can be computed efficiently.

### 3. CRITICALITY PROBABILITIES

Criticality probability is a metric that satisfies the requirements stated above. The criticality probability of an edge is simply defined as the probability of manufacturing a chip in which the critical path includes that edge. Criticality probabilities were first proposed in [7], although in that work *tightness probabilities* were incorrectly assumed to be independent during the computation. A companion paper at this conference [8] describes how these probabilities can be computed efficiently and accurately. The method is briefly reviewed in this section with reference to Fig. 3, and consists of the following steps.

**Step 1** The timing graph is augmented with a “source” and “sink” node, and virtual edges connecting the source node to all primary inputs and the sink node to all primary outputs and timing test nodes. The delay of each input edge is the (statistical) asserted arrival time of the corresponding primary input, and the delay of each output edge is the negative of the (statistical) required arrival time of the corresponding primary output or test point. In the modified graph, the length of the longest path in late mode (i.e., late mode arrival time of the sink node) is quite simply the negative of the late mode slack of the design, while the length of the shortest path (i.e., early mode arrival time of the sink node) is the early mode slack of the design. While this graph augmentation step is not essential in practice, it simplifies the explanation by converting all timing graphs into

Edges for complement slack computation



**Figure 3: Criticality probability computation by the cutset method.**

a single-source single-sink directed acyclic graph. Combinational loops and loops of transparent latches are special cases that require extensions of their deterministic counterpart algorithms to be handled correctly. The following explanation will focus on late mode timing, but analogous arguments can be made for early mode criticality computations.

**Step 2** For a given edge  $e$  of interest, identify any cutset containing  $e$  such that the source and sink node are in opposite partitions. An efficient way of identifying cutsets is suggested in [8].

**Step 3** Compute an *edge slack* for each edge in the cutset. The edge slack of an edge is simply the statistical longest path through that edge. For an edge  $e$  from node  $i$  to  $t$ , the edge slack is simply

$$\text{Edge slack} = AT_i + d_e - RAT_t \quad (1)$$

since  $AT_i$  represents the statistical longest path from the source node to  $i$  and  $-RAT_t$  is the statistical longest path from  $t$  to the sink node.

**Step 4** Compute the *complement edge slack* for edge  $e$ . The complement edge slack is simply the statistical maximum of the edge slack of all edges in the cutset except  $e$ .

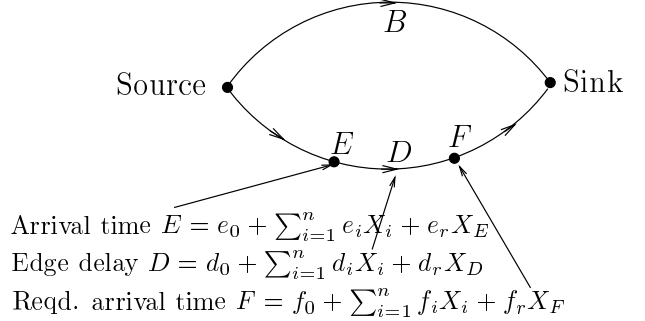
**Step 5** Compute the probability that the edge slack of  $e$  is statistically larger than its complement edge slack. This is the criticality probability of  $e$ .

The reasoning behind the above algorithm is straightforward. Every point in process space has a single critical path at that point, so all we need to do is to find the total criticality probability of all paths that pass through the edge of interest. Every path of the timing graph goes through precisely one edge of the cutset. For this edge to be critical, the *ensemble longest path* through it (i.e., its edge slack) must be larger than all the other edge slacks. Thus the so-called *tightness probability* of the statistical maximum of the edge slack and complement edge slack gives us the criticality probability of the edge. This technique elegantly overcomes previous problems with handling of correlations during criticality analysis. However, this probability still does not tell us how the parametric yield of the chip is impacted by changing the delay characteristics of an edge of the timing graph (by sizing or buffering, for example).

## 4. YIELD GRADIENTS

Our goal is to find the gradient of the parametric yield of the chip to all components of the delay of a single edge of the timing graph. This problem can be divided into two parts. Once we know the statistical slack of the chip as a distribution, the parametric yield is easily expressed in

$$\text{Complementary slack } B = b_0 + \sum_{i=1}^n b_i X_i + b_r X_B$$



**Figure 4: Derivation of yield gradients.**

terms of the CDF of the statistical slack. So if we could isolate the dependence of the statistical slack on the delay of a single edge, we could differentiate that relation to obtain the necessary yield gradients.

Let us focus on a single edge  $D$ . Fig. 4 gives us a convenient abstraction for the derivation. In the figure,  $B$  represents the complement slack of edge  $D$  and comprises the ensemble of all paths of the graph that do not pass through  $D$ . The arrival time  $E$  at the from-node of edge  $D$  represents the delay of the ensemble of paths from the source node to edge  $D$ . Similarly, the required arrival time  $F$  of the to-node of edge  $D$  represents the negative of the delay of the ensemble of paths from edge  $D$  to the sink node. The path Source- $E$ - $D$ - $F$ -Sink is the ensemble of all paths of the timing graph that pass through  $D$ . Thus all paths of the graph have conceptually been split into two buckets – those that pass through  $D$  and those that do not. We have isolated the impact of the edge  $D$  on the statistical slack of the chip.

Next, we can write the statistical slack of the chip as the length of the longest path in Fig. 4

$$\text{Chip slack} \equiv Z = \max(B, E + D - F) \quad (2)$$

where each of  $B$ ,  $E$ ,  $D$  and  $F$  is expressed in first-order canonical form as shown in the figure. The globally correlated process variables are represented by random variables  $X_i, i = 1, 2, \dots, n$  and independently random variation by  $X_B, X_E, X_D$  and  $X_F$ . It is very important to note that  $B$  does not depend on the delay of the edge of interest since it represents the ensemble of paths that *do not pass* through  $D$ . Similarly,  $E$  and  $F$  do not depend on the delay of edge  $D$  since they represent the delay of the longest path in the fanin cone and fanout cone of  $D$ , respectively. Therefore, we can differentiate (2) to obtain the gradient of the chip slack to the components of  $D$ . The rest is simply algebra. To maintain readability of this paper, the differentiation of the max operator by using Clark's formulas [3, 1] has been moved to Appendix A. Using the formulas in the Appendix, yield gradients are derived below in two different scenarios.

### 4.1 Performance maximization at a given yield

This situation is ASIC-like in that we assume a required yield has been specified, and we seek to compute the maximum performance (clock frequency, or, equivalently, slack) at which the chip can safely be operated. Assume that the required parametric yield is  $p$ , a constant. The longest delay of the timing graph in Fig. 4 is a statistical quantity  $Z$

which we assume has a mean value  $z_0$  and a standard deviation  $\sigma_Z$ . Then the corresponding parametric longest path delay that meets the yield requirement  $p$  is denoted by  $z(p)$ . The smaller the value of  $Z$ , the better. The parametric delay which meets the requirement can be expressed as

$$z(p) = \Phi^{-1}(p)\sigma_Z + z_0, \quad (3)$$

where  $\Phi^{-1}()$  is the inverse CDF of a standard normal. This equation simply says that the required yield can be translated into a required number of sigmas on the longest path delay distribution. Our goal now is to obtain the gradient of  $z(p)$  with respect to  $d_0, d_i, i = 1, 2, \dots, n$  and  $d_r$ . This is easily accomplished by simply differentiating (3) with respect to each of these quantities.

$$\frac{\partial z(p)}{\partial d_i} = \Phi^{-1}(p) \frac{\partial \sigma_Z}{\partial d_i} + \frac{\partial z_0}{\partial d_i}, i = 0, 1, \dots, n, r. \quad (4)$$

The two partial derivatives on the right hand side are readily obtained from the boxed equations in the Appendix (24), (30), (31), (33), (35) and (36).

## 4.2 Yield maximization at a given performance

This situation is microprocessor-like in that we assume a target frequency has been specified and we are trying to maximize the yield of chips that will achieve this frequency. Assume that the target frequency translates into a required longest path delay of  $T$  or less. In this case, our parametric yield  $p$  is simply the probability that the longest path  $Z$  has a value that is  $T$  or less. The yield is therefore  $p(T) = \Phi\{(T - z_0)/\sigma_Z\}$ . This simple equation can be differentiated with the help of the formulas in the appendix to obtain the necessary yield gradients

$$\frac{\partial p(T)}{\partial d_i} = \frac{1}{\sigma_Z^2} \phi\left(\frac{T - z_0}{\sigma_Z}\right) \left\{ -\sigma_Z \frac{\partial z_0}{\partial d_i} + (T - z_0) \frac{\partial \sigma_Z}{\partial d_i} \right\} \quad (5)$$

$$i = 0, 1, \dots, n, r.$$

Although the formulas are lengthy, they are easily coded for efficient evaluation on a computer. In terms of implementation, when criticality probability is computed for an edge, a statistical maximum operation is performed between its edge slack and complementary edge slack. At that point, the above formulas allow us to compute yield gradients with respect to individual edge delays.

## 5. APPLICATIONS OF YIELD GRADIENTS

One immediate application of yield gradients is in a formal transistor-sizing optimization context such as EinsTuner [4]. A possible formulation of the optimization problem over the space of transistor widths  $W$  is as follows.

$$\begin{array}{ll} \max_W & \text{parametric yield } p(T) \\ \text{s.t.} & \text{performance requirement } T \\ \text{s.t.} & \text{area, slew and other constraints.} \end{array} \quad (6)$$

In this case, we need to provide to the nonlinear optimizer at each iteration the yield at the required performance  $T$ , and the gradient of that yield with respect to all transistor sizes. This formulation fits well within a transistor-level tuning context since the gradient of the yield with respect to an edge delay can be chain-ruled with the gradient of the edge delay with respect to the individual transistor widths. The latter computation, by the adjoint method, is already a part of the transistor sizing formulation used presently.

Of course, changing a transistor size will typically change the delay of multiple edges of the timing graph as well as fanin edges due to loading effects and fanout edges due to slew effects. All of these sensitivities can be chain-ruled and included in a straightforward manner.

The objective function could easily be changed from parametric yield to *profit* (or any other integral measure of the distribution of the chip slack) in the case of a sorted and speed-binned chip. Sorting implies that different ranges of speeds bring in different amounts of profit. The yield of each bin can be expressed in terms of  $z_0$  and  $\sigma_Z$  as above and then the weighted sum of the profits over all bins differentiated to obtain the necessary gradients.

A second possible formulation is shown below.

$$\begin{array}{ll} \min_W & \text{delay } z(p) \\ \text{s.t.} & \text{a required yield } p \\ \text{s.t.} & \text{area, slew and other constraints.} \end{array} \quad (7)$$

In this formulation, a required yield is specified, and we need to provide to the nonlinear optimizer at each iteration the fastest performance that can be achieved and the gradient of that performance with respect to all transistor widths.

Even in the context of discrete optimization, yield gradients can be used to rank order portions of the circuit that most need attention in order to increase the yield of the circuit to the necessary value, or to increase the performance at a required yield. Having quantitative gradients will decrease the guess-work and try-evaluate-undo loop that is common in physical synthesis during late timing correction and early-mode padding.

Although the actual optimization applications described in this section have not been implemented, the authors feel that yield gradients will enable powerful new ways of optimizing circuits in the presence of variability.

## 6. CONCLUSIONS

Timing closure in the presence of variability implies closure across the entire relevant process space, which is an extremely challenging task. Traditional diagnostics and metrics like critical paths and deterministic timing slacks are inadequate for the optimization task. This paper describes a novel method for computing parametric yield gradients accurately and efficiently, and proposes various applications that can take advantage of yield gradients.

## 7. ACKNOWLEDGMENTS

The authors would like to thank all members of the extended IBM EinsStat and PDS teams for useful discussions and software support.

## 8. REFERENCES

- [1] H. Chang and S. S. Sapatnekar. Statistical timing analysis considering spatial correlations using a single PERT-like traversal. *IEEE International Conference on Computer-Aided Design*, pages 621–625, November 2003. San Jose, CA.
- [2] K. Chopra, S. Shah, A. Srivastava, D. Blaauw, and D. Sylvester. Parametric yield maximization using gate sizing based on efficient statistical power and delay gradient computation. *IEEE International Conference on Computer-Aided Design*, pages 1023–1028, November 2005. San Jose, CA.

- [3] C. E. Clark. The greatest of a finite set of random variables. *Operations Research*, pages 145–162, March–April 1961.
- [4] A. R. Conn, I. M. Elfadel, W. W. Molzen, Jr., P. R. O’Brien, P. N. Strenski, C. Visweswariah, and C. B. Whan. Gradient-based optimization of custom circuits using a static-timing formulation. *Proc. 1999 Design Automation Conference*, pages 452–459, June 1999. New Orleans, LA.
- [5] B. Dodin and S. Elmaghraby. Approximating the criticality indices of the activities in PERT networks. *Management Science*, 31(2):207–223, February 1985.
- [6] X. Li, J. Le, M. Celik, and L. T. Pileggi. Defining statistical sensitivity for timing optimization of logic circuits with large-scale process and enviromental variations. *IEEE International Conference on Computer-Aided Design*, pages 844–851, November 2005. San Jose, CA.
- [7] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, and S. Narayan. First-order incremental block-based statistical timing analysis. *Proc. 2004 Design Automation Conference*, pages 331–336, June 2004. San Diego, CA.
- [8] J. Xiong, V. Zolotov, C. Visweswariah, and N. Venkateswaran. Criticality computation in parameterized statistical timing. *Proc. 2006 TAU (ACM/IEEE workshop on timing issues in the specification and synthesis of digital systems)*, February 2006. San Jose, CA, submitted for review.

## APPENDIX

### A. DIFFERENTIATION OF THE STATISTICAL MAXIMUM OPERATOR

This Appendix derives formulas for differentiation of the statistical max operator with respect to its arguments.

#### A.1 Preliminaries

Let  $A$  and  $B$  be two first-order canonical forms

$$A = a_0 + \sum a_i X_i + a_r X_A, \quad (8)$$

$$B = b_0 + \sum b_i X_i + b_r X_B, \quad (9)$$

where  $X_i$  are the correlated unit-Gaussian random variations,  $X_A$  and  $X_B$  are uncorrelated unit-Gaussian random variations,  $a_i$  and  $b_i$  are sensitivities to correlated random variations, and  $a_r$  and  $b_r$  are the sensitivities to uncorrelated random variations, respectively. The variance of  $A$  and  $B$  and their covariance are

$$\sigma_A^2 = \sum a_i^2 + a_r^2, \quad (10)$$

$$\sigma_B^2 = \sum b_i^2 + b_r^2, \quad (11)$$

$$\text{cov}(A, B) = \sum a_i b_i. \quad (12)$$

Some frequently-used equations are listed below.

$$\phi(r) \equiv \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{r^2}{2}\right) \quad (13)$$

$$\Phi(r) \equiv \int_{-\infty}^r \phi(q) dq \quad (14)$$

$$\theta \equiv (\sigma_A^2 + \sigma_B^2 - 2\text{cov}(A, B))^{1/2} \quad (15)$$

$$\frac{\partial \phi(r)}{\partial r} = -r \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{r^2}{2}\right) = -r \phi(r) \quad (16)$$

$$\frac{\partial \Phi(r)}{\partial r} = \phi(r). \quad (17)$$

Let  $Z = \max(A, B)$ . The mean of  $Z$  is given by

$$\begin{aligned} z_0 &= \Phi\left(\frac{a_0 - b_0}{\theta}\right) a_0 + \left[1 - \Phi\left(\frac{a_0 - b_0}{\theta}\right)\right] b_0 + \theta \phi\left(\frac{a_0 - b_0}{\theta}\right) \\ &= \Phi a_0 + (1 - \Phi) b_0 + \theta \phi. \end{aligned} \quad (18)$$

For simplicity, we have used  $\Phi$  and  $\phi$  to represent  $\Phi\left(\frac{a_0 - b_0}{\theta}\right)$  and  $\phi\left(\frac{a_0 - b_0}{\theta}\right)$ . This notation will be used wherever there is no ambiguity.

The variance of  $Z$  is

$$\sigma_Z^2 = (\sigma_A^2 + a_0^2) \Phi + (\sigma_B^2 + b_0^2) (1 - \Phi) + (a_0 + b_0) \theta \phi - z_0^2. \quad (19)$$

We are interested in the sensitivity of  $z_0$  and  $\sigma_Z$  to the  $a_i$ ,  $b_i$ ,  $a_r$  and  $b_r$  parameters. Because of the symmetry between  $A$  and  $B$ , we will only focus on the derivation of sensitivity with respect to  $a_0$ ,  $a_i$  and  $a_r$ .

### A.2 Sensitivity of mean

#### A.2.1 With respect to mean

We first derive the sensitivity of  $z_0$  with respect to  $a_0$ .

$$\frac{\partial z_0}{\partial a_0} = \frac{\partial \Phi}{\partial a_0} a_0 + \Phi - \frac{\partial \Phi}{\partial a_0} b_0 + \frac{\partial \theta}{\partial a_0} \phi + \theta \frac{\partial \phi}{\partial a_0}. \quad (20)$$

It is easy to show that the following equations hold.

$$\frac{\partial \theta}{\partial a_0} = 0, \quad (21)$$

$$\frac{\partial \Phi}{\partial a_0} = \phi\left(\frac{a_0 - b_0}{\theta}\right) \frac{1}{\theta} = \frac{\phi}{\theta}, \quad \text{and} \quad (22)$$

$$\frac{\partial \phi}{\partial a_0} = -\frac{a_0 - b_0}{\theta} \phi\left(\frac{a_0 - b_0}{\theta}\right) \frac{1}{\theta} = -\frac{a_0 - b_0}{\theta^2} \phi. \quad (23)$$

Therefore, we have

$$\boxed{\frac{\partial z_0}{\partial a_0} = (a_0 - b_0) \frac{\phi}{\theta} + \Phi - \frac{a_0 - b_0}{\theta^2} \phi \theta = \Phi.} \quad (24)$$

#### A.2.2 With respect to correlated sensitivity

Next, we derive the sensitivity of  $z_0$  with respect to the correlated sensitivity term  $a_i$ .

$$\frac{\partial z_0}{\partial a_i} = \frac{\partial \Phi}{\partial a_i} a_0 - \frac{\partial \Phi}{\partial a_i} b_0 + \frac{\partial \theta}{\partial a_i} \phi + \theta \frac{\partial \phi}{\partial a_i}. \quad (25)$$

It is easy to show that the following equations hold.

$$\frac{\partial \Phi}{\partial a_i} = \phi \frac{\partial\left(\frac{a_0 - b_0}{\theta}\right)}{\partial a_i} = -\phi \frac{a_0 - b_0}{\theta^2} \frac{\partial \theta}{\partial a_i}, \quad \text{and} \quad (26)$$

$$\frac{\partial \phi}{\partial a_i} = -\frac{a_0 - b_0}{\theta} \phi \frac{\partial\left(\frac{a_0 - b_0}{\theta}\right)}{\partial a_i} = \frac{(a_0 - b_0)^2}{\theta^3} \phi \frac{\partial \theta}{\partial a_i}. \quad (27)$$

Therefore, we have

$$\begin{aligned}
\frac{\partial z_0}{\partial a_i} &= -\phi \frac{a_0 - b_0}{\theta^2} \frac{\partial \theta}{\partial a_i} (a_0 - b_0) + \frac{\partial \theta}{\partial a_i} \phi + \theta \frac{(a_0 - b_0)^2}{\theta^3} \phi \frac{\partial \theta}{\partial a_i} \\
&= \left\{ -\frac{(a_0 - b_0)^2}{\theta^2} + 1 + \theta \frac{(a_0 - b_0)^2}{\theta^3} \right\} \phi \frac{\partial \theta}{\partial a_i} \\
&= \phi \frac{\partial \theta}{\partial a_i}.
\end{aligned} \tag{28}$$

We need to compute  $\frac{\partial \theta}{\partial a_i}$ , which is given by

$$\begin{aligned}
\frac{\partial \theta}{\partial a_i} &= \frac{\partial(\sigma_A^2 + \sigma_B^2 - 2\text{cov}(A, B))^{1/2}}{\partial a_i} \\
&= \frac{1}{2\theta} \frac{\partial(\sigma_A^2 + \sigma_B^2 - 2\text{cov}(A, B))}{\partial a_i} \\
&= \frac{1}{2\theta} \left\{ \frac{\partial \sigma_A^2}{\partial a_i} + \frac{\partial \sigma_B^2}{\partial a_i} - 2 \frac{\partial \text{cov}(A, B)}{\partial a_i} \right\} \\
&= \frac{1}{2\theta} (2a_i + 0 - 2b_i) \\
&= (a_i - b_i) \frac{1}{\theta}.
\end{aligned} \tag{29}$$

Therefore, in summary, we have

$$\boxed{\frac{\partial z_0}{\partial a_i} = (a_i - b_i) \frac{\phi}{\theta}} \tag{30}$$

### A.2.3 With respect to uncorrelated sensitivity

By similar manipulations, the sensitivity of  $z_0$  with respect to the uncorrelated sensitivity term  $a_r$  is

$$\boxed{\frac{\partial z_0}{\partial a_r} = a_r \frac{\phi}{\theta}} \tag{31}$$

## A.3 Sensitivity of sigma

### A.3.1 With respect to mean

We derive the sensitivity of  $\sigma_Z$  with respect to  $a_0$ .  $\frac{\partial \sigma_Z^2}{\partial a_0}$

$$\begin{aligned}
&= 2\sigma_Z \frac{\partial \sigma_Z}{\partial a_0} \\
&= \left( \frac{\partial \sigma_A^2}{\partial a_0} + \frac{\partial a_0^2}{\partial a_0} \right) \Phi + (\sigma_A^2 + a_0^2) \frac{\partial \Phi}{\partial a_0} - (\sigma_B^2 + b_0^2) \frac{\partial \Phi}{\partial a_0} \\
&\quad + \theta \phi + (a_0 + b_0) \frac{\partial \theta}{\partial a_0} \phi + (a_0 + b_0) \theta \frac{\partial \phi}{\partial a_0} - 2z_0 \frac{\partial z_0}{\partial a_0} \\
&= 2a_0 \Phi + (\sigma_A^2 + a_0^2) \frac{\phi}{\theta} - (\sigma_B^2 + b_0^2) \frac{\phi}{\theta} + \theta \phi \\
&\quad - (a_0 + b_0) \theta (a_0 - b_0) \frac{\phi}{\theta^2} - 2z_0 \Phi \\
&= 2(a_0 - z_0) \Phi + (\sigma_A^2 + a_0^2 - \sigma_B^2 - b_0^2) \frac{\phi}{\theta} - (a_0^2 - b_0^2) \frac{\phi}{\theta} + \theta \phi \\
&= 2(a_0 - z_0) \Phi + (\sigma_A^2 - \sigma_B^2) \frac{\phi}{\theta} + \theta \phi.
\end{aligned} \tag{32}$$

Therefore, in summary, we have

$$\boxed{\frac{\partial \sigma_Z}{\partial a_0} = \frac{1}{2\sigma_Z} \left\{ 2(a_0 - z_0) \Phi + (\sigma_A^2 - \sigma_B^2) \frac{\phi}{\theta} + \theta \phi \right\}} \tag{33}$$

### A.3.2 With respect to correlated sensitivity

We derive the sensitivity of  $\sigma_Z$  with respect to the correlated sensitivity term  $a_i$ . We have  $\frac{\partial \sigma_Z^2}{\partial a_i}$

$$\begin{aligned}
&= 2\sigma_Z \frac{\partial \sigma_Z}{\partial a_i} \\
&= \left( \frac{\partial \sigma_A^2}{\partial a_i} + \frac{\partial a_0^2}{\partial a_i} \right) \Phi + (\sigma_A^2 + a_0^2) \frac{\partial \Phi}{\partial a_i} + \left( \frac{\partial \sigma_B^2}{\partial a_i} + \frac{\partial b_0^2}{\partial a_i} \right) (1 - \Phi) \\
&\quad - (\sigma_B^2 + b_0^2) \frac{\partial \Phi}{\partial a_i} + \left( \frac{\partial a_0}{\partial a_i} + \frac{\partial b_0}{\partial a_i} \right) \theta \phi + (a_0 + b_0) \frac{\partial \theta}{\partial a_i} \phi \\
&\quad + (a_0 + b_0) \theta \frac{\partial \phi}{\partial a_i} - 2z_0 \frac{\partial z_0}{\partial a_i} \\
&= 2a_i \Phi - (\sigma_A^2 + a_0^2 - \sigma_B^2 - b_0^2) \phi \frac{a_0 - b_0}{\theta^2} \frac{\partial \theta}{\partial a_i} \\
&\quad + (a_0 + b_0) \frac{\partial \theta}{\partial a_i} \phi + (a_0 + b_0) \theta \frac{(a_0 - b_0)^2}{\theta^3} \phi \frac{\partial \theta}{\partial a_i} - 2z_0 \frac{\partial z_0}{\partial a_i} \\
&= 2a_i \Phi - 2z_0 \frac{\partial z_0}{\partial a_i} + \left\{ (-\sigma_A^2 - a_0^2 + \sigma_B^2 + b_0^2) \frac{a_0 - b_0}{\theta^2} \right. \\
&\quad \left. + a_0 + b_0 + \frac{(a_0^2 - b_0^2)(a_0 - b_0)}{\theta^2} \right\} \phi \frac{\partial \theta}{\partial a_i} \\
&= 2a_i \Phi - 2z_0 \frac{\partial z_0}{\partial a_i} + \left\{ (-\sigma_A^2 - a_0^2 + \sigma_B^2 + b_0^2 + a_0^2 - b_0^2) \frac{a_0 - b_0}{\theta^2} \right. \\
&\quad \left. + a_0 + b_0 \right\} \phi \frac{\partial \theta}{\partial a_i} \\
&= 2a_i \Phi - 2z_0 \frac{\partial z_0}{\partial a_i} + \left\{ (\sigma_B^2 - \sigma_A^2) \frac{a_0 - b_0}{\theta^2} + a_0 + b_0 \right\} \phi \frac{\partial \theta}{\partial a_i}.
\end{aligned} \tag{34}$$

Therefore, the sensitivity can be computed as  $\frac{\partial \sigma_Z}{\partial a_i}$

$$= \frac{1}{\sigma_Z} \left[ a_i \Phi - z_0 \frac{\partial z_0}{\partial a_i} + \left\{ (\sigma_B^2 - \sigma_A^2) \frac{a_0 - b_0}{\theta^2} + a_0 + b_0 \right\} \phi \frac{\partial \theta}{\partial a_i} \right]$$

$$\boxed{\frac{\partial \sigma_Z}{\partial a_i} = \frac{1}{\sigma_Z} \left[ a_i \Phi - z_0 (a_i - b_i) \frac{\phi}{\theta} + (a_i - b_i) \left\{ a_0 + b_0 + (a_0 - b_0) \frac{\sigma_B^2 - \sigma_A^2}{\theta^2} \right\} \frac{\phi}{2\theta} \right]} \tag{35}$$

### A.3.3 With respect to uncorrelated sensitivity

By similar manipulations, we derive the sensitivity of  $\sigma_Z$  with respect to the uncorrelated term  $a_r$  as

$$\boxed{\frac{\partial \sigma_Z}{\partial a_r} = \frac{1}{\sigma_Z} \left[ a_r \Phi - z_0 a_r \frac{\phi}{\theta} + a_r \left\{ a_0 + b_0 + (a_0 - b_0) \frac{\sigma_B^2 - \sigma_A^2}{\theta^2} \right\} \frac{\phi}{2\theta} \right]} \tag{36}$$