A Rigorous Framework for Convergent Net Weighting Schemes in Timing-Driven Placement *

Tony F. Chan[1,2,3], Jason Cong[1], Eric Radke[2]
[1]Univ. of California, Los Angeles Computer Science Dept.
[2]Univ. of California, Los Angeles Mathematics Dept.
[3]Hong Kong University of Science and Technology
{chan, radke}@math.ucla.edu, cong@cs.ucla.edu

ABSTRACT

We present a rigorous framework that defines a class of *net weighting schemes* in which unconstrained minimization is successively performed on a weighted objective. We show that, provided certain goals are met in the unconstrained minimization, these net weighting schemes are guaranteed to converge to the optimal solution of the original timing-constrained placement problem. These are the first results that provide conditions under which a net weighting scheme will converge to a timing optimal placement. We then identify several weighting schemes that satisfy the given convergence properties and implement them, with promising results: a modification of the weighting scheme given in [11]results in consistently improved delay over the original, 4% on average, without increase in computation time.

1. INTRODUCTION

Approaches for solving the timing-driven placement problem have traditionally been either net-based or path-based; see, e.g., [7] for an overview. *Net weighting* methods, which fall in the latter category, have been a popular tool in analytical placers [16, 8, 9] for handling timing-driven placement. They enjoy a number of advantages, including very low computational complexity and high flexibility. However, net weighting methods suffer the disadvantage that they are largely *ad-hoc*; to date there has been very little theoretical justification for their use [10]. As a result, a number of very different weighting schemes have been proposed, of which some have been shown to be effective in reducing delay. Our goal is to distill the essential properties of a robust and effective net weighting method.

Among commonly used schemes is the *VPR weighting* [11], a *polynomial* scheme defined on the edges by

$$w_e = \left(1 - \frac{\text{slack}(e)}{T_p}\right)^{\alpha} \tag{1.1}$$

where T_p is the max path delay of the previous iterate and α is a user-defined constant. Another scheme is the *PATH weighting* [10], an *exponential* scheme that considers all paths in a circuit efficiently:

$$w_e = \sum_{\pi \ni e} \alpha^{-\frac{\text{slack}(\pi)}{T}} \tag{1.2}$$

where T is a desired max path delay, and α is again a user-defined constant. A third scheme is the *APlace weighting* [9], a *piecewise polynomial* scheme given by

$$w_e = \sum_{\pi \ni e} f\big(\mathit{delay}(\pi), T_u\big), \quad \text{where } f(d, T_u) = \begin{cases} \left(\frac{d}{T_u}\right)^{\alpha} - 1 & \text{if } d > T_u \\ 0 & \text{otherwise} \end{cases} \tag{1.3}$$

Here $T_u = (1-u)T_p$, where u is a constant selected to be 0.1, 0.2, or 0.3, and α is also a user-defined constant.

It has remained an open question under what conditions a net weighting scheme will converge to a timing feasible placement, and what quality can be expected of such a placement. These are addressed in the contributions of this work:

1. We develop a rigorous, generalized framework under which certain net weighting schemes are *guaranteed* to converge to the optimum of the original timing-constrained placement problem, provided the net weighted objective is minimized to a satisfactory degree. We then identify particular net weighting schemes that adhere to this framework.

2. In the case we are able to find global minimizers of unconstrained net weighted objectives, convergence is guaranteed to the global minimizer of the original timing-constrained problem.

3. However, most placers in practice cannot find a global minimizer and instead search for *approximate* local minimizers of the net weighted objectives. In this case, convergence is still guaranteed to a local minimum candidate point of the original timing-constrained problem.

4. We implement convergent weighting schemes in the state-of-the-art placer mPL [4, 12]. When we compare

*This research is partially supported by the National Science Foundation under grants CCF-0430077 and CCF 0528583.

ICCAD'09, November 2-5, 2009, San Jose, California USA.

one scheme, a modification of the VPR weighting, to the original method, we find an average 4% delay improvement on the MCNC benchmarks [18]. In every one of the benchmarks tested, delay improvement was superior for the modified weighting.

The remainder of this paper is organized as follows. In Section 2, we describe the framework and identify net weighting schemes that satisfy the criteria for convergence. In Section 3, we present convergence results when a global minimizer for the net weighted objective can be found. In Section 4, we present convergence results when a local minimizer for the net weighted objective can be approximated. In Section 5, we discuss implementation of the framework and algorithm details. in Section 6 we implement the new schemes in mPL and provide experimental results. In Section 7 the conclusions are summarized and future work is briefly described. Finally, in the Appendix, we present a proof of one of the convergence results.

2. PRELIMINARIES

2.1 Problem Formulation

We use the timing-driven placement problem as the most immediate application of this framework, and refer to it throughout the remainder of this paper. However, the framework need not be restricted to placement, as it can be considered in a general hypergraph-based optimization problem in which net weights are applied to handle timing constraints.

Suppose we wish to minimize total wire length of a circuit by determining the pin locations $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, subject to the constraint that the total delay along any path should not exceed an upper bound $T > 0$. In addition, we consider *generalized* bin density equality constraints $D_r(\mathbf{x}) = 0$ in this formulation. Traditionally, analytical placers (e.g., [15]) divide the placement area into a grid of "bins" and discourage overlapping cells by bounding the average density in each bin using an inequality constraint. In [5], filler "dummy" cells were introduced to convert the inequality constraints into equality constraints and were shown to be effective. The generalized constraints $D_r(\mathbf{x}) = 0$ need not only account for density; in practice, they may account for any equality constraints one may wish to include.

We measure the wire length with the objective function $h_i(\mathbf{x})$ for each net $i = \{i_1, i_2, \ldots, i_m\}$. It is *continuous, nonnegative*, and has the property

$$h_i(\mathbf{x}) = 0 \implies x_{i_1} = x_{i_2} = \cdots = x_{i_m} \tag{2.1}$$

As an example of h_i, we could use the half-perimeter wire length or its log-sum-exp approximation [15, 9]. The quadratic wire length approximation, as in, e.g., [8], is also suitable in this framework.

To measure delay along each edge e in the circuit, we use the *delay function* $d_e(\mathbf{x})$. (Every source-sink pair of pins in a net is considered an edge. We may also consider internal delay of a node by considering an edge between an input and output pin of the node.) The delay function $d_e(\mathbf{x})$ measuring the delay of edge $e = (s, t)$ is *continuous, nonnegative, convex*, and has the property

$$x_s = x_t \implies d_e(\mathbf{x}) = 0 \tag{2.2}$$

For example, the linear and quadratic delay functions, given by $d_e(\mathbf{x}) = \gamma_e |x_t - x_s|$, and $d_e(\mathbf{x}) = \gamma_e (x_t - x_s)^2$, respectively, are suitable in this framework. In the interest of brevity, we often use the *edge delay vector* $\mathbf{d}(\mathbf{x})$, which is defined simply as the vector of delays $d_e(\mathbf{x})$ along each edge e in the circuit.

Further, we consider the *slacks* of the circuit, which are functions of the edge delay vector. For each path π we have the *path slack* σ_π, which is defined in the standard way, as is the *edge slack* σ_e for each edge e; see, e.g., [10] for details. For each net i, we define the *net slack* as $\sigma_i = \min_{e \in i} \sigma_e$.

Given these definitions, we can now formally state the problem we would like to solve:

$$\begin{aligned} &\min_{\mathbf{x}} \sum_{\text{nets } i} h_i(\mathbf{x}) \\ &\textit{subject to:} \\ &\sigma_\pi(\mathbf{d}(\mathbf{x})) \geq 0 \quad \forall \text{ paths } \pi \\ &D_r(\mathbf{x}) = 0 \quad \forall r = 1, \ldots, R \end{aligned} \tag{2.3}$$

Finally, we present a few additional pieces of notation. Let $\mathfrak{F} = \{\mathbf{x} : \sigma_\pi(\mathbf{d}(\mathbf{x})) \geq 0 \ \forall \text{ paths } \pi\}$, and analogously, let $\mathfrak{F}_0 = \{\mathbf{x} : \sigma_\pi(\mathbf{d}(\mathbf{x})) > 0 \ \forall \text{ paths } \pi\}$. Further, let $\mathfrak{D} = \{\mathbf{x} : D_r(\mathbf{x}) = 0, \ r = 1, \ldots, R\}$. We make the mild technical assumptions that $\mathfrak{F}_0 \cap \mathfrak{D} \neq \emptyset$, and that $\mathfrak{F} \cap \mathfrak{D}$ is in the closure of $\mathfrak{F}_0 \cap \mathfrak{D}$.

2.2 Net Weighting Framework

Define the sequence of *weighted objective functions* $N^k(\mathbf{x})$ corresponding to the index $k = 1, 2, \ldots$ as the following:

$$N^k(\mathbf{x}) = \sum_{\text{nets } i} \left[1 + w_i^k(\mathbf{x})\right] h_i(\mathbf{x}) + P^k(\mathbf{x}) \tag{2.4}$$

Here for each net i, $\{w_i^k\}_{k=1}^{\infty}$ is a sequence of *continuous, nonnegative* weighting functions which we require to satisfy the following property, termed the *asymptotic slack control*, for any fixed edge delay vector $\mathbf{d} = \mathbf{d}(\mathbf{x})$:

$$\lim_{k \to \infty} w_i^k(\mathbf{d}) = \begin{cases} 0, & \text{if } \sigma_i(\mathbf{d}) > 0. \\ c_i(\mathbf{d}), & \text{if } \sigma_i(\mathbf{d}) = 0, \text{ where } c_i(\mathbf{d}) \\ & \text{ is some finite constant.} \\ \infty, & \text{if } \sigma_i(\mathbf{d}) < 0. \end{cases}$$

(2.5) Asymptotic slack control.

In (2.4), $P^k(\mathbf{x})$ is a penalty term to handle the generalized density constraints. It is given by

$$P^k(\mathbf{x}) = \mu^k \sum_{r=1}^{R} p_r(D_r(\mathbf{x})) \tag{2.6}$$

Here μ^k is a sequence of positive real numbers such that $\mu^k \nearrow \infty$. The functions p_r are *nonnegative* and have the property

$$p_r(D) = 0 \iff D = 0 \tag{2.7}$$

2.3 Weighting Examples

We now identify particular weighting schemes that satisfy the sufficient criteria for convergence.

We can modify each of the VPR, PATH, and APlace weighting schemes, as defined in (1.1), (1.2), and (1.3), respectively, to satisfy the asymptotic slack control. To do this, we create *sequences* of weighting functions using an increasing *weighting parameter* α_k in place of the ad-hoc, user-defined parameter α. It should be noted that this change to satisfy the asymptotic slack control is necessary for convergence to the optimum: without such a modification, specific circuit examples can be found for which the original VPR, PATH, and APlace methods may fail to arrive at the optimum placement. These examples are simple to construct and are omitted due to space constraints. Also, each of the weighting schemes have been originally defined as *edge* weighting schemes; we broaden their scope to *net* weighting schemes by summing over the edge weights, i.e. $w_i = \sum_{e \in i} w_e$.

We modify the VPR weights to the following, named *VPR-c*:

$$w_i^k(\mathbf{d}(\mathbf{x})) = \sum_{e \in i} \left(1 - \frac{\sigma_e(\mathbf{d}(\mathbf{x}))}{T}\right)^{\alpha_k} \tag{2.8}$$

Here $\{\alpha_k\}$ is a sequence of parameters approaching infinity, and the max delay of the previous placement, T_p, has been replaced with the desired max path delay T. This weighting scheme satisfies the asymptotic slack control.

We use the following modified formulation of the PATH weighting, which we name *PATH-c*:

$$w_i^k(\mathbf{d}(\mathbf{x})) = \sum_{\pi \in \Pi_i} \alpha_k^{-\frac{\sigma_\pi(\mathbf{d}(\mathbf{x}))}{T}} \tag{2.9}$$

Here, Π_i is the set of all paths passing through net i, and $\{\alpha_k\}$ is again a sequence of parameters approaching infinity, with $\alpha_k > 1$ for each k. This weighting scheme satisfies the asymptotic slack control. Figure 2.1 shows the shape of the PATH-c weighting function for increasing values of the net weighting parameter.

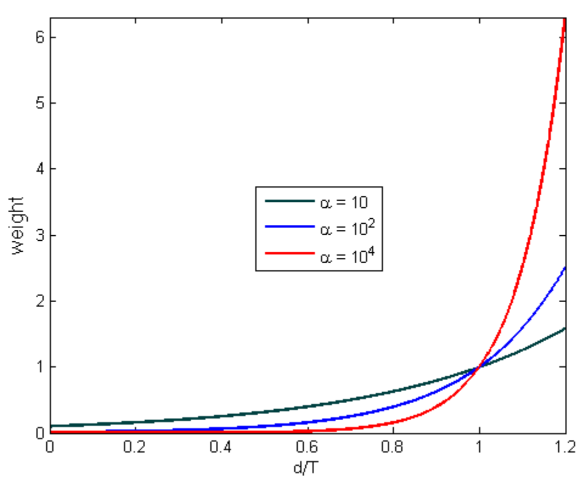

Figure 2.1: PATH-c weighting for increasing net weighting parameter α.

For the APlace weighting, we additionally replace the *desired timing improvement* T_u with the fixed desired max path delay T. The following modified weighting formulation is named *APlace-c*:

$$w_i^k(\mathbf{d}(\mathbf{x})) = \sum_{\pi \in \Pi_i} f^{\alpha_k}(d_\pi(\mathbf{x})),$$
$$\text{where } f^{\alpha}(d) = \begin{cases} \left(\frac{d}{T}\right)^{\alpha} - 1 & \text{if } d > T \\ 0 & \text{otherwise} \end{cases} \tag{2.10}$$

Here, $d_\pi(\mathbf{x}) = \sum_{e \in \pi} d_e(\mathbf{x})$, and $\{\alpha_k\}$ is a sequence of parameters approaching infinity. This weighting scheme satisfies the asymptotic slack control.

Note that the authors of the PATH and APlace weightings have devised efficient methods to calculate the weights, despite the exponential number paths through a net enumerated in their definitions. As described in [10], the PATH algorithm computes all weights in time linear to the number of pins plus the number of edges.

3. GLOBAL CONVERGENCE

Given the above framework using a net weighting objective of the form (2.4), if we can find unconstrained *global* minimizers of the weighted objectives $N^k(\mathbf{x})$, we can guarantee convergence to a *global* minimizer of the original constrained problem.

THEOREM 3.1. *Suppose that $\mathbf{x}^k$ is a global minimizer of the objective $N^k(\mathbf{x})$ in* (2.4) *for each $k = 1, 2, \ldots$. Then every limit point of the sequence $\{\mathbf{x}^k\}$ is a global minimizer of the problem* (2.3).

The proof is a direct extension of that of Theorem A.1 in the Appendix. The general argument is as follows: we first claim that a limit point of $\{\mathbf{x}^k\}$ must be in $\mathfrak{F}$, since otherwise it follows from the *asymptotic slack control* that $\mathbf{x}^k$ will fail to minimize $N^k(\mathbf{x})$ for some sufficiently large k. Then, temporarily assuming the limit point is *not* a global minimizer of the original constrained problem, we show this implies existence of a strictly feasible placement of lower objective value. Taking k sufficiently large and again exploiting the asymptotic slack control, we obtain a contradiction. See [3] for details.

The implication of Theorem 3.1 is that if we can find placements $\mathbf{x}^k$ solving the *unconstrained subproblems* $\min_{\mathbf{x}} N^k(\mathbf{x})$, for $k = 1, 2, \ldots$, the placements $\mathbf{x}^k$ will converge to the globally optimal placement of the constrained problem (2.3).

4. LOCAL CONVERGENCE

4.1 Preliminaries

Given highly non-convex density constraints, finding global minimizers of the weighted objectives N^k is often too difficult to achieve in practice. With this in mind, we now consider the case where the placer can only find a local minimizer for each subproblem $\min_{\mathbf{x}} N^k(\mathbf{x})$ *approximately*. We show that even in this case, the placements will still converge to a Karush-Kuhn-Tucker (KKT) point [13], i.e. a candidate point for a local minimizer, of the following equivalent formulation of the problem (2.3):

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{a}} \sum_{\text{nets } i} h_i(\mathbf{x}) \\ &\textit{subject to:} \\ &a_s + d_e(\mathbf{x}) \le a_t \quad \forall \text{ edges } e, \text{ where } e = (s,t) \\ &0 \le a_j \quad \forall \text{ timing inputs } j \\ &a_\ell \le T \quad \forall \text{ timing outputs } \ell \\ &D_r(\mathbf{x}) = 0 \quad \forall r = 1, \ldots, R \end{aligned} \tag{4.1}$$

Here we introduce the new variable $\mathbf{a} = (a_1, a_2, \ldots, a_n)$, where a_j is an upper bound on the arrival time of pin j.

In addition to the requirements in Section 2, we also require that the weighting w_i^k, delay d_e, net measure h_i, and density penalty p_r are each *continuously differentiable*. The net measure h_i is bounded below by some $\kappa > 0$, and the density penalty p_r additionally is expected to have the property

$$(p_r)'(D) = 0 \implies D = 0$$

We also make some additional requirements about the structure of the weighting functions. First, we require that each w_i^k is *nondecreasing*:

$$\boxed{\nabla_{\mathbf{d}} w_i^k(\mathbf{d}(\mathbf{x})) \geq \mathbf{0}}$$

(4.2) Nondecreasing property.

Thus, increasing any edge delay while holding all other delays constant cannot have the effect of *reducing* a weight.

Secondly, we require that the weighting functions are *non-critically indifferent*: in the limit, changes in non-critical edge delays do not have an impact on the weights.

$$\boxed{\text{If } \sigma_e(\mathbf{d}(\mathbf{x})) > 0, \ \text{ then } \lim_{k\to\infty} \frac{\partial w_i^k(\mathbf{d}(\mathbf{x}))}{\partial d_e} = 0 \ \ \forall \text{ nets } i}$$

(4.3) Non-critical indifference.

Also, we require that each w_i^k is *path consistent*: it can be written as some function of the path delays in the circuit.

$$\boxed{w_i^k(\mathbf{d}(\mathbf{x})) = f_i^k\left(d_{\pi_1}(\mathbf{x}), \ldots, d_{\pi_S}(\mathbf{x})\right)}$$

(4.4) Path consistency.

Finally, we make the mild expectation that the net weighting objectives $N^k(\mathbf{x})$ do not become arbitrarily "jagged" around local minima in $\mathfrak{F}_0$; that is, there exists a $\xi > 0$ independent of k such that if $\mathbf{y} \in \mathfrak{F}_0$ is a local minimum for $N^k(\mathbf{x})$, then $N^k(\mathbf{x})$ is convex in $B_{\mathfrak{F}_0}(\mathbf{y}, \xi) = \{\mathbf{x} : \mathbf{x} \in \mathfrak{F}_0, \|\mathbf{x} - \mathbf{y}\| < \xi\}$.

We can verify that each of the identified convergent weighting schemes VPR-c (2.8), PATH-c (2.9), and APlace-c (2.10) are indeed *nondecreasing*, *non-critically indifferent*, and *path consistent*.

The differentiability requirement is satisfied by the PATH-c weights, and may be satisfied by a slight modification in the VPR-c and APlace-c weights to smooth around non-differentiable points. The non-differentiability is due to the use of max and min functions in computing the slacks in the VPR-c weights, and due to the piecewise nature of the weighting function at the point $d = T$ in the APlace-c weights. The non-differentiable points in the VPR-c weightings can be removed if the log-sum-exp smooth approximations of the max and min functions [15] are used in computing the slacks. The non-differentiable points in the APlace-c weightings can be removed by a simple smoothing of f^a around the point $d = T$.

4.2 Results

We now present the result that if we can find unconstrained *approximate local* minimizers of the weighted objectives $N^k(\mathbf{x})$ (2.4), we can guarantee convergence to a *local* minimizer candidate (KKT point) of the original constrained problem (4.1). A typical technical condition, the *linear independence constraint qualification (LICQ)* [14], is needed at the limit point. It requires that the set of active constraint gradients be linearly independent at the limit point.

THEOREM 4.1. *Suppose that x^k approximates a local minimizer $\mathbf{x}^k_{min}$ of the weighted objective $N^k(\mathbf{x})$ for each $k = 1, 2, \ldots,$ in that $||\mathbf{x}^k - \mathbf{x}^k_{min}|| \to 0$ and $||\nabla N^k(\mathbf{x}^k)|| \to 0$ as $k \to \infty$. If $\mathbf{x}^k \in \mathfrak{F}_0$ for all k sufficiently large, then every limit point of $\{\mathbf{x}^k\}$ at which the LICQ holds is a KKT point of the problem* (4.1).

The KKT conditions are *necessary conditions* for any local minimum $(\bar{\mathbf{x}}, \bar{\mathbf{a}})$ of the problem (4.1). The proof is omitted due to space constraints and may be found in [3]. In the proof, we construct *Lagrange multiplier estimates* and a corresponding value of $\mathbf{a}$ out of every placement $\mathbf{x}^k$. We exploit the structure of the weighting functions to guarantee that each KKT condition is either satisfied at every iteration or in the limit.

The practical implication of Theorem 4.1 is that, given a placer that can find approximate local minimizers for the *unconstrained subproblems* $\min_{\mathbf{x}} N^k(\mathbf{x})$, for $k = 1, 2, \ldots$, we can use the previous placement $\mathbf{x}^{k-1}$ as initial guess for the next unconstrained local minimization. Then the placements $\mathbf{x}^k$ will converge to a very likely local minimum of the constrained problem (4.1). Note that we must ensure that the placements eventually become strictly timing feasible.

We present one final result that relates to the practical implementation of this net weighting framework. Until now, we have made the assumption that the net weights are *continuously* updated according to the current placement $\mathbf{x}$. In practice, we rely on a previous placement $\mathbf{y}$ to compute a set of net weights, then minimize using those fixed weights. Symbolically, we minimize the modified objective function

$$N^k_{\mathbf{d}(\mathbf{y})}(\mathbf{x}) = \sum_{\text{nets } i} \Big(1 + w_i^k(\mathbf{d}(\mathbf{y}))\Big) h_i(\mathbf{x}) + P^k(\mathbf{x})$$

over $\mathbf{x}$. Using the previous iterate $\mathbf{y} = \mathbf{x}^{k-1}$ to set the net weights, we get the following result:

THEOREM 4.2. *Given any $\mathbf{x}^0$, suppose that $\mathbf{x}^k$ is a local minimizer of the weighted objective $N^k_{\mathbf{d}(\mathbf{x}^{k-1})}(\mathbf{x})$ for each $k = 1, 2, \ldots,$ that $||\mathbf{x}^k - \mathbf{x}^{k-1}|| \to 0$, and that $\mathbf{x}^k \in \mathfrak{F}_0$ for all k sufficiently large. Then $\{\mathbf{x}^k\}$ converges to a KKT point of the problem* (4.1)*, provided that point satisfies the LICQ.*

The proof is omitted due to space constraints; see [3]. We discuss the practical implications of these results below in Section 5.

5. IMPLEMENTATION

Motivated by the results in Sections 3 and 4, to solve the problem (4.1), we can iteratively perform unconstrained minimization of $N^k_{\mathbf{d}(\mathbf{x}^{k-1})}(\mathbf{x})$ using the previous solution $\mathbf{x} \leftarrow \mathbf{x}^{k-1}$ as a starting point. In each outer iteration, we set $\mathbf{x}^k \leftarrow \mathbf{x}$ if $\mathbf{x}$ approximates a local minimizer within some tolerance τ^k, where $\tau^k \searrow 0$. This framework is described in Algorithm 5.1.

We implement each of the weighted objectives described in Section 2.3 in the state-of-the-art placer mPL [5]. Although significantly more sophisticated than the example given in

Algorithm 5.1 Example, iterative placement with net weighting.

Choose initial placement $\mathbf{x}^0$, tolerances $\{\tau^k\}$ such that $\tau^k \searrow 0$, and convergence tolerance $\rho > 0$.
$k \leftarrow 0$
while $k \leq 1$ or $||\mathbf{x}^k - \mathbf{x}^{k-1}|| > \rho$ **do**
  $k \leftarrow k + 1$
  $\mathbf{x} \leftarrow \mathbf{x}^{k-1}$
  while $||\nabla N^k_{\mathbf{d}(\mathbf{x}^{k-1})}(\mathbf{x})|| > \tau^k$ **do**
    iterate $\mathbf{x}$ in unconstrained minimization of $N^k_{\mathbf{d}(\mathbf{x}^{k-1})}(\mathbf{x})$
  end while
  $\mathbf{x}^k \leftarrow \mathbf{x}$
end while

Algorithm 5.1, similar principles are followed in mPL. We update the weighting parameter once every outer loop iteration, using the previous iterate to calculate net weights, then solve the inner loop subproblems using the Uzawa algorithm [1] for smoothed density equality constraints. A decreasing schedule of tolerances for cell overlap is used to determine sufficient convergence of the iterates. We use the log-sum-exp wire length approximation [15, 9] and a quadratic delay model, which satisfy all conditions for h_i and d_e required in this framework.

6. EXPERIMENTAL RESULTS

In Table 6.1, we measure the efficacy of each method implemented in mPL on selected MCNC benchmarks [18]. These circuits were used because they contained the necessary timing information; the ISPD '05 and '06 contest examples could not be used due to lack of such information. The Cadence RTL compiler was used to synthesize the circuits with the Nangate 45nm open cell library. The results represent the best result obtained for each scheme in 8 runs, measured by shortest max delay after detailed placement; statistics are given for the placement after both global and detailed placement. The net weights were updated once every outer iteration in the mPL placer, which occurred approximately 50-70 times before sufficient convergence was attained. Note that net weightings were only applied during the global placement phase, so some degradation in the delays can be observed after the global placement phase. Column "WL" gives the half-perimeter wire length, "Del." gives the max path delay of the circuit, and "CPU" is a measure of the computation time necessary to complete the placement. The values are scaled against those obtained using the regular, non-weighted mPL placer. On average, the VPR-c scheme yields the best improvement in delay, the smallest increase in CPU time compared to non-weighted mPL, and matches the APlace-c weights in smallest increase in wire length compared to non-weighted mPL.

In Table 6.2, we compare the VPR-c weighting method against the original method in [11], which we term "VPR." The difference between the two schemes is that the VPR scheme is flat, without increasing net weighting parameter, and the max delay is set dynamically to be the current max delay at every static timing analysis (VPR). The entries in the table are arranged as those in Table 6.1, with the exception that computation time has been omitted, as it did not vary significantly between the two methods (static timing analysis and re-calculation of net weights are carried out at every outer iteration in both methods). As in Table 6.1, the best result over 8 runs for each method is shown, measured by the shortest max delay after detailed placement. After detailed placement, the VPR-c scheme nearly matched the VPR scheme in wire length while outperforming it in delay on all 10 benchmarks, 4% on average. Thus, the modifications necessary for theoretical convergence yield an improvement over the method in [11].

Table 6.1: Comparison of weighting schemes in mPL.

		Global		Detailed		
Circuit	Weight	WL	Del.	WL	Del.	CPU
ex5p	VPR-c	1.14	0.86	1.04	0.88	1.61
	PATH-c	1.07	0.95	1.01	0.93	2.16
	APlace-c	1.13	0.88	1.03	0.89	3.81
alu4	VPR-c	1.02	0.82	1.02	0.85	1.31
	PATH-c	1.19	0.92	1.10	0.93	2.16
	APlace-c	1.02	0.95	1.01	0.95	1.93
apex2	VPR-c	1.05	0.85	1.02	0.89	1.49
	PATH-c	1.17	0.98	1.12	0.96	1.77
	APlace-c	1.12	0.83	1.03	0.91	2.26
pdc	VPR-c	1.09	0.80	1.03	0.82	1.48
	PATH-c	1.11	0.89	1.06	0.88	1.81
	APlace-c	1.08	0.78	1.03	0.82	7.22
apex4	VPR-c	1.04	0.87	1.01	0.95	1.42
	PATH-c	1.10	0.88	1.06	0.91	1.72
	APlace-c	1.02	0.88	1.00	0.97	2.47
des	VPR-c	1.02	0.91	1.00	0.93	1.29
	PATH-c	1.05	0.95	1.03	0.98	1.48
	APlace-c	1.01	0.92	1.00	0.93	2.64
ex1010	VPR-c	1.07	0.82	1.03	0.87	1.34
	PATH-c	1.08	0.84	1.06	0.85	1.72
	APlace-c	1.05	0.86	1.02	0.89	2.52
misex3	VPR-c	1.07	0.85	1.02	0.95	1.42
	PATH-c	1.05	0.92	1.03	0.96	1.63
	APlace-c	1.08	0.82	1.03	0.93	2.09
seq	VPR-c	1.04	0.82	1.03	0.84	1.32
	PATH-c	1.11	0.84	1.08	0.87	1.55
	APlace-c	1.05	0.81	1.03	0.86	1.83
spla	VPR-c	1.01	0.84	1.01	0.90	1.32
	PATH-c	1.16	0.84	1.13	0.85	1.70
	APlace-c	1.04	0.85	1.01	0.91	5.17
Ave.	VPR-c	1.05	0.85	1.02	0.89	1.40
	PATH-c	1.11	0.90	1.07	0.91	1.77
	APlace-c	1.06	0.86	1.02	0.91	3.19

7. CONCLUSIONS AND FUTURE WORK

In this work, we determined a set of criteria defining a class of net weighting schemes that were shown to converge to optimal placements for the original timing-constrained problem. When a *global* minimizer to the unconstrained weighted objective can be found, the essential property for convergence of a weighting algorithm is the *asymptotic slack control.* In the more practical case when only an *approximate local* minimizer to the unconstrained weighted objective can be found, a weighting must also be *nondecreasing, non-critically indifferent,* and *path consistent.* Several schemes satisfying these properties were identified and im-

Table 6.2: Comparison of VPR-like methods in mPL.

		Global		Detailed	
Circuit	Weight	WL	Del.	WL	Del.
ex5p	VPR-c	1.04	0.90	1.01	0.90
	VPR	1.01	0.93	1.01	0.93
alu4	VPR-c	1.02	0.82	1.02	0.85
	VPR	1.01	0.91	1.02	0.92
apex2	VPR-c	1.05	0.85	1.02	0.89
	VPR	1.01	0.95	1.02	0.95
pdc	VPR-c	1.09	0.80	1.03	0.82
	VPR	1.04	0.87	1.01	0.87
apex4	VPR-c	1.04	0.87	1.01	0.95
	VPR	1.01	0.87	1.01	0.96
des	VPR-c	1.02	0.91	1.00	0.93
	VPR	1.03	0.91	1.02	0.94
ex1010	VPR-c	1.07	0.82	1.03	0.87
	VPR	1.01	0.88	1.01	0.91
misex3	VPR-c	1.07	0.85	1.02	0.95
	VPR	1.01	0.89	1.01	0.97
seq	VPR-c	1.04	0.82	1.03	0.84
	VPR	1.01	0.89	1.01	0.89
spla	VPR-c	1.01	0.84	1.01	0.90
	VPR	1.00	0.92	1.00	0.91
Ave.	VPR-c	1.05	0.85	1.02	0.89
	VPR	1.01	0.90	1.01	0.93

plemented in the mPL placer on the MCNC benchmarks. The *VPR-c* scheme outperformed all others, improving delay by an average of 11% while increasing wire length by 2% and increasing computation time by 40%, as compared to unweighted placement. Further, VPR-c outperformed the original VPR weighting scheme with improved delay in all MCNC benchmarks tested, an average of 4%; additionally, minimal increase in wire length and no change in computation time was observed.

In the future, we plan to implement Lagrangian schemes for comparison with our net weighting methodology. It can be shown that, using a scheme with projection of the multipliers as in [6], the method can be viewed as a net weighting that satisfies many of the required properties in the framework. We would like to further investigate this and the potential insights it can provide. In addition, we plan to extend the net weighting methodology to handle the timing-driven sizing and placement problem.

8. ACKNOWLEDGMENTS

The authors gratefully acknowledge Guojie Luo and Kirill Minkovich of the UCLA Computer Science Department, John Lee and Prof. Lieven Vandenberghe of the UCLA Electrical Engineering Department, and the anonymous reviewers for their valuable comments, assistance, and insightful discussions.

APPENDIX

A. GLOBAL CONVERGENCE PROOF

Here we prove the first main result of this work, Theorem 3.1. In order to do so, we introduce necessary additional notation, which is presented in Table A.1 for reference.

Table A.1: Notation for the Appendix.

Symbol	Definition
$\mathcal{N}$	The (finite) set of all nets in the circuit
$\mathcal{P}$	The (finite) set of all pins in the circuit
$\mathcal{P}_\mathcal{I}$	The set of all input pins in the circuit
$\mathcal{P}_\mathcal{O}$	The set of all output pins in the circuit
$\mathcal{P}_\mathcal{M}$	$\mathcal{P} \setminus (\mathcal{P}_\mathcal{I} \cup \mathcal{P}_\mathcal{O})$
$A_t(\mathbf{d}(\mathbf{x}))$	Arrival time of pin t (see [10])
$R_s(\mathbf{d}(\mathbf{x}))$	Required arrival time of pin s (see [10])
Π	The set of all paths in the circuit
$\mathcal{E}$	The set of all edges in the circuit
M	The number of edges in the circuit
$G(\mathbf{x})$	$\sum_{i\in\mathcal{N}} h_i(\mathbf{x})$
$\Psi^k(\mathbf{x})$	$\sum_{i\in\mathcal{N}} w_i^k(\mathbf{d}(\mathbf{x}))h_i(\mathbf{x})$

Note that we require that $\mathcal{P}_\mathcal{I} \cap \mathcal{P}_\mathcal{O} = \emptyset$, i.e. there are no single-pin paths.

As handling of the penalty term $P^k(\mathbf{x})$ can be done via standard techniques (as in, e.g., [13]), we ease the discussion by omitting the generalized density constraints in the timing-constrained problem and simply assuming that some pins are fixed instead. The analogous problem to (2.3) that we will solve is:

$$\begin{aligned} &\min_{\mathbf{x}} \ G(\mathbf{x}) \\ &\textit{subject to:} \\ &\sigma_\pi(\mathbf{d}(\mathbf{x})) \geq 0 \quad \forall \text{ paths } \pi \in \Pi \\ &\text{(some pins fixed)} \end{aligned} \tag{A.1}$$

As there are no density constraints to handle in this formulation, the net weighted objective becomes

$$N^k(\mathbf{x}) = \sum_{\text{nets } i} \left[1 + w_i^k(\mathbf{x})\right] h_i(\mathbf{x}) = G(\mathbf{x}) + \Psi^k(\mathbf{x})$$

THEOREM A.1. *Suppose that $\mathbf{x}^k$ is a global minimizer of the weighted objective $N^k(\mathbf{x})$ for each $k = 1, 2, \ldots$. Then every limit point of the sequence $\{\mathbf{x}^k\}$ is a global minimizer of the problem (A.1).*

Before proving Theorem A.1, we present the following lemma and corollary. The proof is omitted due to space constraints; see [3] for full details.

LEMMA A.2. *For any edge delay vector $\mathbf{d} = [d_1 \ \ldots \ d_M]^T$,*

$$\sigma_e(\mathbf{d}) = \min_{\pi \ni e}\{\sigma_\pi(\mathbf{d})\}$$

COROLLARY A.3. *The condition*

$$\lim_{k\to\infty} w_i^k(\mathbf{d}) = \begin{cases} 0, & \text{if } \sigma_\pi(\mathbf{d}) > 0 \quad \forall \pi \in \Pi_i. \\ c_i(\mathbf{d}), & \text{if } \sigma_\pi(\mathbf{d}) \geq 0 \quad \forall \pi \in \Pi_i. \\ & (c_i(\mathbf{d}) \text{ is a finite const.}) \\ \infty, & \text{if } \exists \pi \in \Pi_i \ \textit{s.t.} \ \sigma_\pi(\mathbf{d}) < 0. \end{cases} \tag{A.2}$$

is equivalent to the asymptotic slack control (2.5).

Now we are ready for the proof of Theorem A.1.

PROOF. Let $\mathbf{z}$ be a global minimizer of the problem (A.1). First note that since each $\mathbf{x}^k$ minimizes $N^k(\mathbf{x})$, we have that

$$N^k(\mathbf{x}^k) \leq N^k(\mathbf{z}), \text{ for all } k$$

i.e.,

$$G(\mathbf{x}^k) + \Psi^k(\mathbf{x}^k) \leq G(\mathbf{z}) + \Psi^k(\mathbf{z}), \text{ for all } k \qquad \text{(A.3)}$$

Suppose $\mathbf{x}^*$ is a limit point of $\{\mathbf{x}^k\}$, so there exists some subsequence K such that $\lim_{k\in K} \mathbf{x}^k = \mathbf{x}^*$.

We first show that $\mathbf{x}^* \in \mathfrak{F}$. We have by continuity that $\sigma_\pi(\mathbf{d}(\mathbf{x}^*)) = \lim_{k\in K} \sigma_\pi(\mathbf{d}(\mathbf{x^k}))$ for each path π. Suppose that $\sigma_\pi(\mathbf{d}(\mathbf{x}^*)) < 0$ for some π. Then there exists some S such that for all $k \in K$ with $k > S$,

$$\sigma_\pi(\mathbf{d}(\mathbf{x^k})) < -\epsilon$$

for some $\epsilon > 0$. By property (A.2) it follows that for each net i through which π passes,

$$w_i^k(\mathbf{d}(\mathbf{x}^k)) \to \infty$$

Furthermore, since $\sigma_\pi(\mathbf{d}(\mathbf{x}^*)) < 0$ it follows by definition that $\sum_{e\in\pi} d_e(\mathbf{x}^*) > T > 0$, so there exists at least one $\hat{e} = (\hat{s}, \hat{t}) \in \pi$ such that $d_{\hat{e}}(\mathbf{x}^*) > 0$. Thus by (2.2), $x^*_{\hat{s}} \neq x^*_{\hat{t}}$ so that for the net $\hat{i} \ni \hat{e}$, it follows from (2.1) and nonnegativity that $h_{\hat{i}}(\mathbf{x}^*) > 0$. Then again by continuity we can choose $k \in K$ sufficiently large so that for some $R > 0$,

$$h_{\hat{i}}(\mathbf{x}^k) > R$$

Thus we get:

$$\begin{aligned} \Psi^k(\mathbf{x}^k) &= \sum_{\text{nets } i\in\mathcal{N}} w_i^k(\mathbf{d}(\mathbf{x}^k))h_i(\mathbf{x}^k) \\ &\geq w_{\hat{i}}^k(\mathbf{d}(\mathbf{x}^k))h_{\hat{i}}(\mathbf{x}^k) \\ &\geq w_{\hat{i}}^k(\mathbf{d}(\mathbf{x}^k))R \to \infty \text{ as } k \to \infty \end{aligned} \qquad \text{(A.4)}$$

Now, since $\mathbf{z} \in \mathfrak{F}$, it follows from (A.2) that $\lim_{k\in K} \Psi^k(\mathbf{z}) = C$ for some constant $C \geq 0$. This fact, combined with (A.4), contradicts (A.3) for sufficiently large $k \in K$. We thus conclude that $\sigma_\pi(\mathbf{d}(\mathbf{x}^*)) \geq 0$ for all paths π, so that $\mathbf{x}^* \in \mathfrak{F}$.

Now suppose that $\mathbf{x}^*$ is *not* a global minimizer of (A.1), i.e. $G(\mathbf{x}^*) > G(\mathbf{z})$. We will show that this implies that there exists a point $\mathbf{y} \in \mathfrak{F}_0$ such that $G(\mathbf{x}^*) > G(\mathbf{y})$.

If $\mathbf{z} \in \mathfrak{F}_0$, take $\mathbf{y} = \mathbf{z}$. Otherwise, choose any point $\tilde{\mathbf{x}} \in \mathfrak{F}_0$, and assume $G(\mathbf{x}^*) \leq G(\tilde{\mathbf{x}})$ (otherwise, take $\mathbf{y} = \tilde{\mathbf{x}}$). Now let $\mathbf{x}_\beta = \beta\tilde{\mathbf{x}} + (1-\beta)\mathbf{z}$ for $\beta \in (0, 1)$. For each path π, we know that $\sigma_\pi(\mathbf{d}(\tilde{\mathbf{x}})) > 0$ and $\sigma_\pi(\mathbf{d}(\mathbf{z})) \geq 0$. It follows by convexity of the delay function that the pathwise slack function is concave; hence $\sigma_\pi(\mathbf{d}(\mathbf{x}_\beta)) \geq \beta\big(\sigma_\pi(\mathbf{d}(\tilde{\mathbf{x}}))\big) > 0$, so that $\mathbf{x}_\beta \in \mathfrak{F}_0$ for all $\beta \in (0, 1)$.

Now it follows by continuity that we can choose $\beta = \beta_1$ sufficiently small so that $G(\mathbf{x}_{\beta_1}) < G(\mathbf{x}^*)$. We have found our desired point $\mathbf{y} = \mathbf{x}_{\beta_1}$.

Now recall that by definition of $\mathbf{x}^k$,

$$G(\mathbf{x}^k) + \Psi^k(\mathbf{x}^k) \leq G(\mathbf{y}) + \Psi^k(\mathbf{y}), \text{ for all } k$$

We can take the limit of both sides of this inequality to obtain

$$G(\mathbf{x}^*) + \lim_{k\in K} \Psi^k(\mathbf{x}^k) \leq G(\mathbf{y})$$

and so by nonnegativity of each w_i^k and h_i,

$$G(\mathbf{x}^*) \leq G(\mathbf{y}) \qquad \text{(A.5)}$$

which contradicts our previous assertion that $G(\mathbf{x}^*) > G(\mathbf{y})$. We conclude that $\mathbf{x}^*$ is a global minimizer of (A.1). $\square$

B. REFERENCES

[1] K. Arrow, L. Huriwcz and H. Uzawa, *Studies in Nonlinear Programming*, Stanford University Press, Stanford, CA, 1958.

[2] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.

[3] T. F. Chan, J. Cong, and E. Radke, “Convergent net weighting schemes in hypergraph-based optimization,” tech. report, UCLA Computational and Applied Math Reports, 2009.

[4] T. F. Chan, J. Cong, J. R. Shinnerl, K. Sze, and M. Xie, “mPL6: enhanced multilevel mixed-size placement,” *Proc. 2006 Int’l Symp. on Phys. Design*, pp. 212-214, 2006.

[5] T. F. Chan, J. Cong, and K. Sze, “Multilevel generalized force-directed method for circuit placement,” *Proc. 2005 Int’l Symp. on Phys. Design*, pp. 185-192, April 2005.

[6] C.-P. Chen, C. C. N. Chu, and D. F. Wong, “Fast and exact simultaneous gate and wire sizing by Lagrangian relaxation,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 18, No. 7, pp. 1014-1025, July 1999.

[7] J. Cong , T. Kong , J. Shinnerl , M. Xie, and X. Yuan, “Large-scale circuit placement,” *ACM Trans. Des. Automat. Electron. Syst.*, vol. 10, no. 2, pp. 389-430, Apr. 2005.

[8] H. Eisenmann and F. M. Johannes, “Generic global placement and floorplanning,” *Proc. 35th ACM/IEEE Design Automation Conference*, pp. 269-274, 1998.

[9] A. B. Kahng and Q. Wang, “Implementation and extensibility of an analytic placer,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 24, No. 5, pp. 1-14, May 2005.

[10] T. Kong, “A novel net weighting algorithm for timing-driven placement,” *Proc. 2002 IEEE/ACM Int’l Conf. on Computer-aided Design*, pp. 172-176.

[11] A. Marquardt, V. Betz, and J. Rose, “Timing-driven placement for FPGAs,” *Proc. 2000 ACM/SIGDA Eighth Int’l Symp. on Field Programmable Gate Arrays*, pp. 203-213, February 2000.

[12] G.-J. Nam and J. Cong (eds.), *Modern Circuit Placement: Best Practices and Results*, Springer, 2007.

[13] S. G. Nash and A. Sofer. *Linear and Nonlinear Programming*, McGraw-Hill, 1996.

[14] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, New York, NY, 1999, pp. 490-525.

[15] W. Naylor, R. Donelly, and L. Sha, “Non-linear optimization system and method for wire length and delay optimization for an automatic electronic circuit placer,” US Patent 6671859, 2003.

[16] R. S. Tsay and J. Koehl, “An analytic net weighting approach for performance optimization in circuit placement,” *Proc. Design Automation Conf.*, pp. 620-625, 1991.

[17] M. H. Wright, “Interior methods for constrained optimization,” *Acta Numerica 1992*, pp. 341-407.

[18] S. Yang, “Logic synthesis and optimization benchmarks, version 3.0,” tech. report, Microelectronics Center of North Carolina, Research Triangle Park, NC, 1991.