# Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 04-035

## Implicit Heuristics to Mitigate Interconnect Congestion in a Multilevel Placement Framework

Navaratnasothie Selvakkumaran, Phiroze Parakh, Abhishek Ranjan,
and George Karypis

September 29, 2004

# Implicit Heuristics to Mitigate Interconnect Congestion in a Multilevel Placement Framework

Navaratnasothie Selvakkumaran Phiroze Parakh Abhishek Ranjan

George Karypis

September 28, 2004

**Abstract**

The congestion minimization techniques have become more important due to the shrinking geometries and "taller" interconnects, causing numerous design convergence problems. Also, multilevel placement algorithms are becoming more prevalant due to their ability to natively incorporate mixed-mode placement, in addition to their ability to scale to very large design sizes. In this context, we have developed a number of implicit heuristics for minimizing congestion in the process of fortifying an existing industrial multilevel placment tool (Dolphin). In contrast to the explicit congestion heuristics that explicity measure congestion either by stochastic estimators or by approximate global routing, our techniques primarily rely on pre-emptively identifying congestion prone clusters and making amends to them. Essentially, we intervene during the clustering phase of the multilevel placement to identify such congestion prone clusters and try to increase the supply of routing resources to those clusters. Increasing the supply of routing resources can be done by whitespace injection. Cell/cluster inflation is however, not a new technique, but what is new in our techniques is that we inflate the clusters *before* any placement information is obtained. In addition to the effective schemes of cluster inflation that reduce congestion substantially (upto 25% on average), we have also modified the clustering formulation to generate clusters that are less prone to congestion. These new clustering schemes do not use any additional area and as a result a more attractive option for designs with very high utilization. The non-use of addtional whitespace does not mean they are ineffective, even though these formulations are dwarfed in quality by whitespace techniques, they reduce congestion at a significant rate of 20% on average. Furthermore, a

new area distribution heuristic is also developed that reduce congestion by 32% on average, and by far this is the best among the techniques develped.

Part of the success of our schemes is derived from the novel metric we used to identify clusters with higher congestion risk. We first describe well-known implicit congestion metrics such as pin-density and bin-degree, and then examine the pros and cons of deploying them for our purpose. The new metric called "perimeter-degree" is developed to over come the disadvantages of pin-density and bin-degree in our multilevel placement setup.

Due to the simplicity of the heuristics presented, the run time penalty is extremely small. Not only that, but these techniques can also be applied in a complementary fashion to the existing explicit congestion mitigating methods, which further improves the desirability of these techniques.

# 1 Introduction

As the technology progresses, the distance between interconnects on the chip layout decreases, which leads to coupling induced timing closure problems and signal integrity issues. The problem is excerbated in the congested areas of the chip in addition to routability problems traditionally encountered in these areas. This paper looks at a totally new avenue to address for interconnect congestion. Specifically, we present clustering techniques developed in a multilevel placement context, that reduces the congestion without incurring substantial computational cost. These techniques were developed in 2002 with the aid of Dolphin [1] RTL to GDSII physical implementation tool of Monterey Design Systems Inc. Some of the results of this paper had been published previously in System Level Interconnect Prediction (SLIP) Workshop (2003) [23].

## 1.1 Existing Work on Mitigating The Congestion

Routability is a fundamental requirement of any placement engine. A placement engine is tasked to produce layouts that are routable, while using minimal routing resources (wirelength and via count). However, there is no known formulation for expressing and subsequently optimizing for routability within a placement engine. Optimizing for wirelength is the closest acceptable alternative for routability [5, 27]. By minimizing wirelength, we reduce the lower bound for rectilinear Steiner-tree length of the nets in the design, which leads to globally reduced total routing demand.

The two common methods for minimizing wirelenth are analytical schemes [6], [26] and hypergraph partitioning driven placement methods [4, 28, 32, 35].

However, concentrating solely on wirelength can produce densely connected local regions that lead to congestion. A number of techniques have been developed to cope with and mitigate the congestion. These techniques in general broadly follow the steps of generating a placement, measuring the congestion, identifying congested regions and then mitigating for it. An augmented analytical formulation for spreading the cells apart in congested regions is provided in [11]. An alternative strategy involves reducing the capacity utilization bounds of the congested regions by allocating white space to them [21, 30]. A similar intuition is used in two other schemes [3, 13], where instead of allocating the white space to regions, the cells lying in the congested regions are inflated. After these adjustments, the placement for the affected regions are recomputed. A different approach is proposed in [14], where local nets are redistributed to minimize a routability metric while attempting to maintain lower wirelength through a multi-objective simulated-annealing based algorithm. This approach was presented as a post-process to global placement, perhaps due to its high computational complexity. In addition, larger proportion of the congestion reduction reported in [14] may have been purely due to wirelength reduction as the results presented show substantial reduction in placement wirelength. A multi-constraint partitioning driven placement tool is advocated in [34] that reduces the congestion by reducing the maximum pin density and the maximum bin degree.

## 1.2 The Methods of Measuring Interconnect Congestion

Most physical design tools depend on global router to estimate the congestion. In [21] authors use a regional router for estimating congestion. A fast incremental global router coupled with physical hierarchy generation heuristic is used for congestion estimation in [7]. As an alternative to computationally expensive global routers, much faster stochastic congestion estimating techniques have emerged recently. These techniques [15, 19] with reasonably high fidelity identify congested regions by assigning probabilities for each possible path a net could take and then summing up these regional probabilistic values. A similar global net-length distribution prediction model for heterogeneous systems is presented in [33].

Another approach to measure the regional congestion effortlessly is by looking at the regional wirelength. An

extensive literature exists in estimating and measuring these wirelength distributions with the help of regional Rent's exponent [9, 10, 24]. The basic intuition behind these schemes rely on the fact that the regional Rent's exponent of the congested regions tends to be higher than the rest of the regions. The regional Rent's exponent also known as *local Rent's exponent* can be measured from the local distribution of number of interconnects and their lengths [20]. A partitioner derived local Rent's exponent estimate is used for regional average wirelength estimation in [31].

For simplicity we can categorize the congestion into two, *inter congestion* and *intra congestion*. The inter congestion is caused by the global interconnects connecting regions, and the intra congestion denotes the congestion caused by interconnects completely hidden inside regions. In a multilevel placement framework, the intra congestion at a coarser level will manifest itself as the inter congestion at a finer level, as the nets hidden inside clusters/regions at a coarser level become exposed in one of the finer levels. For this reason, we are primarily concerned with the inter congestion in this paper. Simple metrics to measure the inter congestion and the intra congestion are bin degree (number of nets exposed from a bin) and pin density, respectively.

## 1.3 Our Contributions

We present simple intuitive schemes that augment the coarsening phase of the multilevel placement. By leaving the uncoarsening phase as it is, the existing refinement heuristics can be used without any modifications. This allows the placement algorithm to scale to large design sizes and as the experimental results would show, effeciently reduces the congestion substantially. The techniques presented in this paper are complimentary to other well-known explicit techniques, in the sense, both can be deployed together to futher reduce congestion. Due to the simplicity of the techniques presented, they can be integrated into any multilevel placement algorithm easily.

The Section 2 defines various terms and notations used in the paper and the characteristics of the multilevel placement engine Dolphin ([1]) used for our experimentation. Section 3 explains the problem, selection of implicit congestion metrics and the reasons for choosing them. In Section 4, we describe the heuristics developed and then in Section 5 we provide empirical validation of the same. Finally, concluding remarks are given in Section 6.

## 2 Notations and Background

A *netlist* $G = (V, E)$ is a set of cells $V$ and a set of nets $E$. Each net is a subset of the set of cells $V$. The *size* of the net is the cardinality of this subset. A cell $v$ is said to be *incident* on a net $e$, if $v \in e$. The number of unique nets incident on a vertex $v$ is said to be its *degree* and denoted by $d(v)$. Similar to the definition of the degree of the cell, cluster degree and bin degree are defined to be the number of unique nets incident on the cluster and the bin respectively. Each cell $v$ and net $e$ has a weight associated with them and they are denoted by $w(v)$ and $w(e)$, respectively. The placement problem is to find $(x, y)$ coordinates for all cells such that they do not overlap in a chip. One of the objectives of the placement is to minimize the half-perimeter wirelength. The half-perimeter wirelength of a net is the half the perimeter of the smallest rectangle formed to encompass all the incident cells of that particular net. The half-perimeter wirelength of the placement essentially sums up the half-perimeters of all nets present.. A *pin* is an interface between any cell $v$ and any net $e$.

The multilevel methodology used in placement consists of three distinct stages; *coarsening*, *finding initial solution* and *uncoarsening*. In the coarsening stage, the netlist is successively *clustered* to create a series of coarser netlists such that each coarser netlist represent successively smaller version of the original problem. A *cluster* is a subset of cells merged together for the purpose of enabling the creation of coarser netlist. Thus finding the placement of cluster of a coarser netlist is approximately equivalent to finding the placement of cells of the finer netlist contained within the cluster, although further legalisation and refinement may be required for the finer netlist. The second stage of finding initial solution achieves an initial placement of the coarsest (smallest) netlist. In the third stage, this initial solution is propagated through progressively finer netlists and refined using any iterative optimization scheme. The book *Multilevel Optimization in VLSI CAD* [8] provides more detailed descriptions of different multilevel optimization strategies used in VLSI CAD.

The sections of Dolphin tool [1], most relevant for this paper involves a multilevel global placement engine implemented in the following way. First, the netlist is successively clustered by randomly choosing a cell/cluster and merging it with one or more cells/clusters adjacent to it. The choice of adjacent cells/clusters is determined by a *group average connectivity* metric. The value of this metric for two clusters $u$ and $v$ is shown by Equation 1.

$$connectivity(u,v) = \sum_{\forall e\{e|e\in E \wedge e\in u \wedge e\in v\}} \frac{w(e)}{w(u)\times w(v)\times(|e|-1)} \quad (1)$$

Then the coarsest netlist is placed on $k$ bins using a prototyping floorplanner. This placement is called *level 0* placement. The uncoarsening stage involves quadrisecting the $4^l k$ bins of level $l$ to obtain the $4^{l+1}k$ bins of level $l+1$. A 4-way extension of Fiduccia-Mattheyses (FM) [12] algorithm implemented similar to the way suggested by [25, 22], is used for quadisectioning as well as for further refinement of a particular level. This algorithm directly minimizes half-perimeter wirelength. During the uncoarsening process, the netlist is also uncoarsened in parallel to quadrisectioning of bins and as a result finer and finer levels of netlist are used successively. The uncoarsening process stops when there are 10-30 cells in each bin and no more finer netlist remains to be placed. Let this last placement level be $m$-th level placement, and it contains $4^m k$ bins.

Furthermore, we use the global router of the Dolphin [1], to measure the effectiveness of our schemes. We do not use the post detail route congestion estimate, as the layer assignments may skew the results. Note, during the regular operation, the placement engine is made aware of layers, and produces placement results that show high correlation between post global route congestion and post detail route congestion. Since we use the simple wirelength objective, we restrict ourselves to the congestion estimates obtained after the global routing. The congestion at the global routing level is reported in terms of the number of congested route edges. These route edges are categorized into "severe congested edges" whose routing demand exceeds 110% of the routing supply and "mild congested edges" whose routing demand is between 100% and 110% of the routing supply. We also report routed wirelength as a sum of global routed tree lengths.

# 3 Motivation

In this section we describe the motivation behind developing implicit heuristics. First we describe the reasons for developing this set of heuristics and the project constraints. Later, we identify metrics that can help us to develop implicit and the motivation for choosing them.

## 3.1 Project Definition

The congestion is a local phenomenon, which occurs when routing demand exceeds available routing resources of a local region. The routing demand consists of two components; the *global demand*, which is caused by wires that pass over the region, and the *local demand*, which primarily originate from a local region. To estimate the first component directly, either global routers or stochastic estimators are required, therefore we do not intend to measure global demand explicitly, as our design goal is to measure and account for the congestion implicitly. The second component, local demand could potentially be measured easily by a plethora of metrics such as the number of exposed nets (bin degree), pin density, local wirelength, local Rent's exponent etc. One of the factors that determine the variation in global demand is the location of a region. The central regions of the chip will have more global demand due to the 2D nature of the chip [27], and also the corners of mega cells are well-known trouble spots. Another factor that causes changes in global demand variation is the local demand of the adjacent regions. That is a region sandwiched between two regions with high local demand will experience higher global demand, as more "over pass" wires go through that region to connect the adjacent high demand regions. Therefore by focusing on local demand we can reduce both the maximum local demand and the peak location-independent global demand. The heuristics presented in this paper were developed with the intention of reining on peak local demand.

During the development of these heuristics, we were constrained to develop implicit heuristics. The reasons for this state of affairs was twofold. Firstly, the existing tool Dolphin [1] was quite capable of explicitly measuring and accounting for the congestion. Secondly, the multi-objective formulations similar to [14, 34] could not be used as the requirement of this project required us to not to modify the objective function of the refinment algorithm. As a result, we used the regular half-perimeter minimizing objective as our standard objective to evaluate relative performances of our heuristics. Note, that these implicit schemes presented here cause least disruption to incorporate in the existing tool, and more importantly they can be deployed in a complementary fashion to the existing explicit congestion minimizing schemes.

## 3.2 Metrics that Affect Congestion

The following sections describe the metrics that can be efficiently measured for identifying congestion and investigates inter relationships among them. We first discuss the importance of implicit metrics such as pin-density and bin-degree in identifying congestion. These two metrics are well-known [34]. We estimated the correlation between pin-density and congestion as well as the correlation between bin-degree and congestion. This analysis of large number of benchmarks show that both pin-density and degree show a positivie correlation value of about 0.7 on average to the congestion measured after global routing.

In the subsequent sections, we describe the limitation of degree in identifying routing demand in clusters with varying sizes, and then develop a new metric called "perimeter-degree" to effectively identify potential congestion in clusters.

### 3.2.1 Pin Density

One of the best known implicit parameters that affect the congestion distribution is the distribution of pin-density. Since each pin is required to be connected to a route track, regions with high pin-density typically suffer from higher congestion. Similar to the behaviour of any density metric, pin-density also follows Observation 1. This simple observation can be easily leveraged to provide bounds on pin-density of the placement. For example, if the maximum possible pin-density allowed any where on the chip is $\eta$, then we could easily analyse the cells and for all cells whose pin-density exceeds $\eta$ we could pad them with white space, so that the pin-density of none of the cell exceeds $\eta$. Ofcourse, this assumes that a certain amount of white space is available in the design. Modern designs typically contain large amounts of white space, so instead of distributing it equally or distributing it for the purpose of relaxing the partitioning balance bounds, we could also use it to pad the cells with higher risk of congestion in advance to avoid congestion hotspots implicitly.

**Observation 1.** *The maximum pin-density of the cells within a cluster is the upper bound of the pin-density of that cluster. Similarly, the maximum pin-density of any cell or cluster present in the design will be the maximum possible pin-density for any bin or region of the chip.*

### 3.2.2 Degree

The degree of a cluster plays a unique role in determining the congestion. When all clusters contain single cells, the degree is equivalent to pin count. However, for clusters containing more cells, degree captures routing demand more accurately than pin count. This discrepancy is primarily brought about by the clustering choices made in a multilevel setup. For example, in Figure 1 three different clusterings are shown. Each of these three clusters ($a$,$b$,$c$) show merger of two sub clusters of equal area and pin count (hence equal pin-density). The figure also shows 7,8 and 7 nets involved in the merger of clusters $a$,$b$ and $c$ respectively. Cluster $a$ shows three hidden 2-pin nets while cluster b shows two hidden 2-pin nets, and the third cluster $c$ shows some higher cardinality nets. Each exposed net require atleast a single routing track to complete the connection between the cells inside a particular cluster and the cells outside of it. Thus, to route the clusters $a$,$b$ and $c$ we would in the minimum require 4,6 and 7 tracks respectively. These numbers correspond to the degree of these clusters. From this Observation 2 follows.

**Observation 2.** *The minimum number of routing tracks required to route the cells inside a cluster/bin to the rest of the clusters/bins is equivalent to its degree.*

From this observation the intuition behind minimizing the maximum degree of the bins or clusters as one of the strategies to lower congestion should be clear. One might raise the problem of ignoring the interior of clusters/bins for estimating congestion. But remember, the reason we need to focus on the degree of the cluster is due to the use of multilevel paradigm, where the hidden interior of the clusters ought to be handled at a level finer than the level these clusters are handled. Nets of the interiors of the clusters usually become exposed nets (degree) of finer clusters, thus degree is a sufficiently good metric to handle congestion in a multilevel framework. Before we proceed to the next discussion, we would also like to emphasize the limitation of pin count in expressing the routing demand of the clusters in Figure 1. These clusters demonstrate the need to consider the cluster degree in properly expressing routing demand.

The degree of a bin also plays a major role in determining the wirelength associated with that bin. The wirelength is the prime factor of congestion [27]. By a similar argument, we could say that the local regions with high wirelength are highly likely to be congested. This becomes clear, if we consider the fact that a particular local region would have

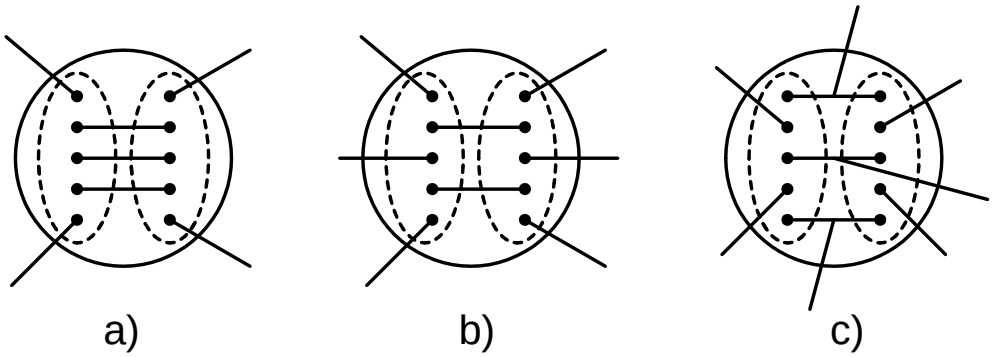


Figure 1: Three examples of cluster formation.

required a seperate chip on a technology node just a few generations ago and the total wirelength of that separate chip would largely correspond to a local wirelength of that paricular region of the current generation chip. However, the relationship between the degree and the associated wirelength was previously undetected. We can infer the relationship between the degree and the wirelength from the relationship between number of interconnects and the lengths of interconnects described in [20]. In [20] authors show that when the number of interconnects and their length of a local region are plotted, the gradient of the fitted line corresponds to $2r^l - 3$, where $r^l$ is local Rent's exponent. From the regular Rent's formula [18] ($P = kB^r$, where $P$ - degree, $B$ - number of blocks, $k$ - constant, $r$ - Rent's exponent), we know that the Rent's exponent ($r$) increases when degree of a region ($P$) increases. Thus, when degree ($P$) increases, local Rent's exponent $r^l$ increases, which result in the fitted line with the gradient $2r^l - 3$, becoming more horizontal. This implies that as the number of interconnects increases, the the length of the newly added interconnects tends to be longer than the existing lengths. In other words, the total length of the interconnects is increasingly dominated by the newly added longer interconnects. Coupled with this phenomenon, the fact that each unit increase in the degree requires an addition of atleast a single a new interconnect leads to a previously unrecognized relationship between degree and local wirelength (Observation 3). It might also be of interest to note that the multi-objective formulation presented in [14], used an exponentially penalizing cost for higher bin degree with the intention of balancing the bin degree.

**Observation 3.** *A linear increase in the degree of a local region such as a bin will super linearly increase the wirelength associated with that bin.*

At this juncture, it is relevant to point out another analysis done to identify the importance of the bin degree. For this analysis, two sets of placements were generated, the first set with the wirelength objective only and the second set using both the wirelength objective and the congestion objective (explicit techniques available in Dolphin). The latter set of placements were on average less congested in terms of 50% reduction in "severe congested edges", 18% reduction in "mild congested edges" and 2% reduction in routed wirelength. As expected, the maximum bin degree of the second set compared with the first was lower, and on average it was found to be lower by 14%. This clearly illustrates the importance of degree in controlling the interconnect congestion.

However, as you may have noticed already, the degree represents only the demand for routing resources. But the congestion is the discrepancy between demand and supply of the routing resources. We know that pin count represents routing demand and the pin-density represents both demand and supply and hence pin-density is a suitable metric for estimating congestion. Similarly, we need to identify a degree related metric that can be used to estimate congestion. Note, when the area of all regions involved are same, such as bins, then degree alone is a sufficient indicator of high congestion, because routing supply available to equal sized areas is approximately same (In this paper, we ignore location dependent variations to supply). However, for clusters this is not true as the clusters often vary in their sizes, even though the connectivity formula shown in Equation 1, is said to encourage relatively equal sized clusters. One possible degree related metric coined in the same fashion as pin-density, could be obtained by dividing the degree by the area of the cluster. However, this formulation assumes the supply of the "interior" of the clusters too, which need to be handled at a finer cluster level and not at the current cluster level. Therefore, we need to consider the supply at the periphery of the clusters. Degree is a measure of routing demand at the periphery of the cluster, therefore it is prudent to have an estimate that measures supply also at the periphery of the cluster. We could visualize the supply of the cluster as the number of routing tracks incident on the cluster, had it been placed on the chip. If we know the Rent's exponent of the layout of the routing tracks, we can estimate this accurately, but as we found empirically, a simple metric of supply being proportional to the perimeter suffices. Again, we are developing an implicit scheme that need to operate on the clusters before the placement information is known. This does not allow us to determine the shape of the cluster for the purpose of measuring perimeter, nevertheless, an estimate of the perimeter as the square root of the area turned out to be a good estimate. Note, this estimate of the routing supply corresponds to the Rent's exponent

value of $0.5$. This new degree related congestion metric, named as *"perimeter-degree"* is defined by Equation 2. Variants of this metric obtained by changing the supply Rent's exponent from 0.5 to values in between 0.5 and 0.7, did not empirically produce superior result in our experimental setup. These variants and the degree related metric obtained by dividing the degree of cluster by the area of the cluster, might however be suitable for other tools and/or benchmarks.

$$perimeter - degree(u) = \frac{d(u)}{\sqrt{w(u)}} \tag{2}$$

# 4 Implicit Techniques for Congestion Mitigation

In this section, we present three categories of implicit congestion mitigation techniques. The first category of techniques essentially pad white space to the cells/clusters that are deemed to have higher risk of being placed on a congested region. The second category of techniques alters the clustering formulations of the multilevel placement. In the third category, we present a novel area re-distribution scheme which was empirically found to be superior to the other schemes.

## 4.1 White Space Aided Techniques

The intuition for the white space aided techniques is simple, we first identify a set of cells/clusters that have higher likelihood of being placed in a congested region and then allocate white space to the cells/clusters before they are placed. As these inflated cells/clusters take up more space of the chip, they also receive more supply of routing tracks, thus reducing the congestion. The idea of adding white space to congested regions is not a new one, but to the best of our knowledge no previous work has focused on pre-emptively allocating white spaces to the cells/clusters.

As we discussed earlier Section 3.2.1, we could also merely look at the pin-density of the cells and inflate the cells until all cells obey the desired upperbound for pin-density of the placement $\eta$. But instead of pin-density we use "perimeter-degree" as it is a more appropriate metric for identifying congestion in clusters for the reasons discussed in Section 3.2.2.

Essentially, these techniques first estimate the perimeter-degree and then set a limit on the perimeter-degree of the cells/clusters. For all cells/clusters exceeding this limit, the algorithm adds pseudo area (white space) in proportion to the deviation from this limit. Note, that the denominator of metric perimeter-degree is a square root of area (Equation 2), hence increasing the area will decrease the perimeter-degree of the cell/cluster.

In order to understand the effect of padding white space to cells/clusters we present the techniques separately for cells, clusters and then for both. These padding of white space is done as part of the clustering phase of the multilevel global placement. The cells/clusters for padding white space is determined by the statistical parameters mean $\mu$ and standard deviation $\sigma$ of perimeter-degree values of all movable cells. The cells/clusters exceeding a certain threshold value determined by a function of statistical parameters, are inflated in proportion to the deviations of the perimeter-degree values from the threshold perimeter-degree values. Macro nodes are however not considered as they are treated as fixed nodes in our experimental setup. Subsequent to the global placement, the cells with inflated areas are restored with their original areas, so that the rest of the physical design steps proceed with actual area. In the subsequent subsections we provide variations of the techniques described here.

#### 4.1.1 Inflating Cells

In this experimental setup we modify the original netlist by inflating the cells with high perimeter-degree before the multilevel placement is initiated. In order to take advantage of varying utilization factors of the designs involved, we set three different threshold limits based on the utilization factors of the design. For benchmarks with less than 60% utilization, we use the threshold of $\mu - 0.5\sigma$, for designs with utilization between 60% and 80%, we use the mean $\mu$ as the threshold, and for the rest we use $\mu + 0.5\sigma$ as our threshold. Then for each of the movable cells $u$ with perimeter-degree values larger than the threshold limit, its new area is set as $w(u)/(threshold-limit/perimeter-degree(u))$. Note that these threshold values are chosen somewhat arbitrarily, and empirically we found that their utilization factors did not exceed 14%. These threshold values can be tuned to obtain the desired trade-off between decrease in congestion and the increase in utilization. Our intention is to empirically demonstrate the potential of using perimeter-degree as the congestion metric and the viability of pre-emptively padding white space.

#### 4.1.2 Inflating Clusters

In the previous scheme, the cells of the original netlist were inflated, but the clusters of the intermediate netlists were not perturbed. In this experiment we inflate in the clusters of the intermediate netlists only by identifying the clusters with relatively higher perimeter-degree values and pad them with additional area. Inflating the cells of the original netlist "spreads out" the inherently dense portions of the netlist. However, during the creation of coarser netlist new densely connected regions appear due to the use of congestion-unaware clustering formulation. Therefore, in order to spread out such newly formed densely connected regions we inflate the clusters. Similar to cell inflation idea, we pad additional area to the clusters with high congestion. Each cluster $u$ with perimeter-degree higher than the threshold of $\mu + \sigma$ are set with new area of $w(u)/((\mu + \sigma)/perimeter - degree(u))$.

Although inflating clusters after the clusters are formed diverges from orignal clustering objective (Equation 1) or constraint used, it captures and addresses interconnect density variations introduced by the clustering formulation (for example as in Figure 1 ).

#### 4.1.3 Inflating Both Cells and Clusters

It is also possible to inflate both the cells and the clusters. In this method, we combine the heuristics of both Section 4.1.1 and Section 4.1.2, by inflating all the cells that have perimeter-degree above $\mu + 0.25\sigma$, and dynamically inflating all the clusters that have perimeter-degree above $\mu + 1.75\sigma$. We empirically found these threshold values to be appropriate for our tool set.

### 4.2 New Clustering Techniques

We present two clustering formulations intended to reduce the number of clusters formed with relatively higher perimeter-degree. First scheme modifies the connectivity formula, while the second one uses a novel cluster filtering based technique. Both these schemes *do not* pad additional area to cells/clusters and hence will remain effective even in high utilization designs.

#### 4.2.1 Perimeter-degree Constrained Connectivity

During the coarsening phase of multilevel partitioning algorithms, a sequence of successively smaller netlists is constructed by finding groups of vertices and merging them together to form the vertices of the next level coarser netlist. A number of schemes have been developed for selecting what groups of vertices will be merged together to form single vertices in the next level coarse netlists [17, 16, 2, 29]. These schemes are essentially variations of the basic edge connectivity metric [16] shown in Equation 3.

$$edge - connectivity(u, v) = \sum_{\forall e\{e|e \in E \wedge e \in u \wedge e \in v\}} \frac{w(e)}{(|e| - 1)} \tag{3}$$

From this edge connectivity estimation, our default connectivity Equation 1 was derived for the purpose of limiting the formation of large clusters. The denominator value of $w(u) \times w(v)$ of Equation 1, is used to discourage the formation of clusters with large areas. Motivated by this behaviour, we transformed our default connectivity function Equation 1, to a new formulation that uses the perimeter-degree of the clusters instead of the area of the clusters as part of the denominator function, as shown in Equation 4. The intuition behind this new formulation is to discourage the merger of clusters with higher congestion risk.

$$new - connectivity(u, v) = \sum_{\forall e\{e|e \in E \wedge e \in u \wedge e \in v\}} \frac{w(e)}{perimeter - degree(u) \times perimeter - degree(v) \times (|e| - 1)} \tag{4}$$

#### 4.2.2 Filtering Clusters

During the coarsening stage, a randomly selected cell is always merged with one of its neighbors, irrespective of the merge quality, although neighbors are chosen in a locally greedy fashion. In the new clustering technique, we first chose more candidate pairs than the required number of pairs for merger, then we selectively merge only a subset of them. To determine the desirability of the pairs of clusters we evaluate them in terms of the perimeter-degree of the resulting clusters. We do not complete the merger of pairs, but estimate the perimeter-degree of the resulting cluster

from the parameters of the pair of clusters such as degree, area and the number of common nets. In our scheme, we typically chose five times as many as pairs as needed and then discard all the pairs that have perimeter-degree values larger than the average perimeter-degree value. From the remaining pairs, we chose the pairs for merger in the order of priority determined by Equation 1.

### 4.3 Area Re-distribution Based Technique

In this heuristic, we let the placement tool use the actual area of cells for the first few placement levels and then intervene to change the area of the cells in proportion to their perimeter-degree, such that total area of cells remain unchanged. This area re-distribution scheme achieves two objectives, first by relying on the actual area of the cells for the placement of first few levels, the purturbations are limited to the finer sized bins, and secondly by allocating area in proportion to the perimeter-degree value, all the clusters have similar congestion risk. More specifically, for the levels $m-2$,$m-1$ and $m$, we re-distribute the area of the clusters according to their perimeter-degree value, such that total area of clusters remain the same.The intuition behind this scheme is that the clusters with higher congestion risk will become larger and hence increase their supply to reduce congestion risk, and this additional area is "borrowed" from clusters that are less likely to get congested. We achive this in the following manner. First we calculate the perimeter-degree values and then multiply that with $total-area-of-cells/total-perimeter-degree$ to obtain the pseudo area values. Once areas of the clusters are re-distributed as such the new perimeter-degree values of all clusters would be equal. We let the default legalization routines to handle any overflows that may occur, but we did not encounter any noteworthy violations.

## 5 Experimental Validation

We experimentally validated our heuristics against our default placement obtained purely by minimizing half-perimeter wirelength as described in Section 2. For this purpose, we used a suite of 15 industrial benchmarks, details of which are listed in Table 1

The results are presented as tables of ratios of "RouteWL", "PlaceWL", "No. Mild" and "No. Severe". RouteWL

| | $\|V\|$ | $\|E\|$ | # macros | Util. | No. of Route Edges | | |
|---|---|---|---|---|---|---|---|
| | | | | | Total | Mild | Severe |
| m01 | 57638 | 60933 | 13 | 39.9 | 635116 | 25832 | 6887 |
| m02 | 100245 | 103404 | 33 | 42.3 | 1263091 | 7639 | 110 |
| m03 | 22336 | 27449 | 8 | 39.1 | 200810 | 27931 | 6055 |
| m04 | 22830 | 23041 | 13 | 50.3 | 94181 | 4190 | 195 |
| m05 | 153263 | 201186 | 33 | 76.2 | 307562 | 45603 | 6940 |
| m06 | 78767 | 78522 | 0 | 69.8 | 155263 | 40073 | 9796 |
| m07 | 152178 | 195139 | 10 | 77.1 | 262267 | 138972 | 67353 |
| m08 | 117413 | 121239 | 54 | 79.7 | 625575 | 25936 | 2545 |
| m09 | 16056 | 18291 | 0 | 70.1 | 52812 | 19068 | 9177 |
| m10 | 99276 | 111610 | 0 | 65.8 | 189043 | 156690 | 129957 |
| m11 | 25409 | 29844 | 0 | 85.9 | 58789 | 13050 | 1628 |
| m12 | 56083 | 62745 | 4 | 81.5 | 83905 | 29299 | 11539 |
| m13 | 40730 | 45351 | 0 | 82.9 | 42953 | 5802 | 439 |
| m14 | 45369 | 45309 | 0 | 88.1 | 333166 | 4416 | 168 |
| m15 | 262079 | 327060 | 12 | 85.1 | 602795 | 21612 | 2894 |

Table 1: The characteristics of the benchmarks used. The column labeled as "Total" represents the total number of edges of the routing grid. The columns labeled as "Mild" and "Severe" indicate the number of mild congested edges and number of severe congested edges of the default placement, respectively.

is the total tree length of all the routed nets. PlaceWL measures the half-perimeter placement wirelength. "No. Mild" and "No. Severe" respectively indicate the number of mildly congested edges and the number of severely congested edges. All these measurements are average results of three different runs. The ratios were obtained by dividing the metric of the new placement by the metric of the default placement. Thus, when the ratios are larger than 1.0, that means that particular metric is more in the new placement compared to the default placement. Therefore, values greater than 1.0 are considered degradations and values less than 1.0 are considered improvements.

Finally, the last column of the tables present the geometric mean of the ratios and it helps to compare the overall impact of each scheme. Note, that the reason we use geometric mean of ratios instead of mean of ratios is the ability of the geometric mean to give equal importance to comparable degradations and improvements For example, an average of two ratios representing twice as good (0.5) and twice as bad (2.0) would result in 1.25 and be interpreted as 25% worse, but the geometric mean of same ratios (0.5 and 2.0) would result in 1.0 and properly show that each of these two ratios cancel each other out.

## 5.1 Empirical Evaluation of White Space Aided Techniques

In the following sections we provide the empirical evaluation of white space aided techniques presented in Section 4.1.

| | m01 | m02 | m03 | m04 | m05 | m06 | m07 | m08 | m09 | m10 | m11 | m12 | m13 | m14 | m15 | Geo.Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RouteWL | 1.03 | 1.01 | 1.00 | 0.99 | 0.98 | 0.99 | 0.94 | 1.00 | 0.99 | 1.01 | 1.00 | 1.00 | 1.00 | 1.01 | 1.02 | 0.998 |
| PlaceWL | 1.03 | 1.01 | 1.01 | 1.00 | 0.98 | 0.99 | 1.01 | 1.00 | 1.01 | 1.00 | 1.00 | 1.00 | 1.00 | 1.01 | 1.02 | 1.005 |
| No. Mild | 0.78 | 0.63 | 0.89 | 0.66 | 0.96 | 0.97 | 0.89 | 0.99 | 0.99 | 1.00 | 0.96 | 0.99 | 1.03 | 1.02 | 1.07 | 0.912 |
| No. Severe | 0.46 | 0.55 | 0.92 | 0.13 | 0.90 | 0.78 | 1.04 | 1.01 | 0.92 | 0.98 | 0.99 | 0.99 | 1.05 | 0.93 | 1.16 | 0.778 |

Table 2: The results of cell inflation in terms of routed wirelength, place wirelength, the number of mildly congested edges and the number of severely congested edges are compared against the default placement and provided here as ratios. The last column provides the geometric mean of these ratios.

| | m01 | m02 | m03 | m04 | m05 | m06 | m07 | m08 | m09 | m10 | m11 | m12 | m13 | m14 | m15 | Geo.Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RouteWL | 1.10 | 1.02 | 1.02 | 1.00 | 0.99 | 1.00 | 0.98 | 0.99 | 0.99 | 0.99 | 0.99 | 1.00 | 1.01 | 1.01 | 1.01 | 1.007 |
| PlaceWL | 1.07 | 1.02 | 1.02 | 1.01 | 0.99 | 1.00 | 1.04 | 1.00 | 1.01 | 1.00 | 0.99 | 1.00 | 1.01 | 1.01 | 1.02 | 1.012 |
| No. Mild | 1.19 | 0.78 | 1.02 | 0.73 | 0.95 | 0.95 | 0.96 | 0.89 | 0.98 | 1.01 | 1.11 | 1.02 | 0.98 | 1.14 | 0.86 | 0.964 |
| No. Severe | 1.30 | 0.46 | 1.09 | 0.17 | 0.67 | 0.84 | 1.06 | 0.79 | 0.94 | 1.01 | 1.43 | 1.01 | 0.60 | 0.90 | 0.68 | 0.784 |

Table 3: The results of cluster inflation in terms of routed wirelength, place wirelength, the number of mildly congested edges and the number of severely congested edges are compared against the default placement and provided here as ratios. The last column provides the geometric mean of these ratios.

#### 5.1.1 Inflating Cells

TheTable 2 illustrates the effectiveness of the cell inflation heuristic presented in Section 4.1.1. In this technique cells with high-perimeter degree are inflated before the placement starts. The number of severe congested edges were 16% worse in one benchmark (m15), but upto 97% better in another (m04). The overall PlaceWL and RouteWL show no changes, but the number of mild congested edges and the number of severe congested edges are on average lower by 9% and 22% respectively.

#### 5.1.2 Inflating Clusters

The idea behind inflating clusters is to account for interconnect density variations introduced by the clustering as well as the inherent variations present in the design as described in Section 4.1.2. The results of this heuristic are illustrated in Table 3. The number of severe congested edges were 43% worse in m11 but upto 93% better in another benchmark (m04). But in the overall, the number of severe congested edges were 22% better on average, similarly the the number of mild congested edges were better by 4%. Both PlaceWL and RouteWL has slipped slightly by about 1%.

| | m01 | m02 | m03 | m04 | m05 | m06 | m07 | m08 | m09 | m10 | m11 | m12 | m13 | m14 | m15 | Geo.Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RouteWL | 1.04 | 1.00 | 1.01 | 0.99 | 0.99 | 1.01 | 0.96 | 1.01 | 1.01 | 1.00 | 1.01 | 1.00 | 1.01 | 1.03 | 1.00 | 1.005 |
| PlaceWL | 1.03 | 1.00 | 1.02 | 1.00 | 0.99 | 1.01 | 1.02 | 1.01 | 1.02 | 1.00 | 1.01 | 0.99 | 1.02 | 1.03 | 1.00 | 1.010 |
| No. Mild | 1.01 | 0.80 | 0.87 | 0.85 | 0.93 | 1.02 | 0.86 | 1.03 | 1.00 | 1.00 | 0.87 | 0.99 | 1.00 | 0.99 | 0.90 | 0.938 |
| No. Severe | 0.62 | 0.61 | 0.79 | 0.29 | 0.70 | 0.98 | 1.03 | 1.05 | 1.00 | 1.00 | 0.45 | 0.89 | 1.34 | 0.49 | 0.90 | 0.756 |

Table 4: The results of inflating both cells and clusters in terms of routed wirelength, place wirelength, the number of mildly congested edges and the number of severely congested edges are compared against the default placement and provided here as ratios. The last column provides the geometric mean of these ratios.

#### 5.1.3 Inflating Cells and Clusters

This scheme combines the ideas behind previous two schemes as described in Section 4.1.3. The results are presented in Table 4. The PlaceWL slips by about 1% but RouteWL is almost the same. The number of severe congested edges were 34% worse in one benchmark (m13) but in the overall 24% better on average. The number of mild congested edges were reduced by 6%.

### 5.2 Empirical Evaluation of New Clustering Techniques

This section provides the empirical evaluation of new clustering techniques developed in Section 4.2. Notice, that both of the techniques developed does not alter the utilization and hence suitable even for very high utilization designs.

#### 5.2.1 Perimeter-degree Constrained Connectivity

The evaluation of the new clustering formulation intended to limit pairing up of clusters with relatively large perimeter-degree as described in Section 4.2.1 is presented in Table 5. One of the significant observations is the reduction in both PlaceWL and RouteWL. If we consider the fact that very high effort is always spent to reduce the wirelength, then this reduction in wirelength of approximately 4% clearly shows the superiority of the new formulation in enabling the refinement algorithm to find better solution consistently. This added benefit of wirelength reduction does not however diminish the importance of the new formulation in reducing congestion, as the number of mild congested edges and the number of severe congested edges were also reduced by 13% and 20% respectively.

| | m01 | m02 | m03 | m04 | m05 | m06 | m07 | m08 | m09 | m10 | m11 | m12 | m13 | m14 | m15 | Geo.Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RouteWL | 0.89 | 0.95 | 1.00 | 0.97 | 0.94 | 0.97 | 0.93 | 0.99 | 0.99 | 0.93 | 0.99 | 0.98 | 0.95 | 0.98 | 0.91 | 0.959 |
| PlaceWL | 0.89 | 0.95 | 1.00 | 0.97 | 0.94 | 0.97 | 0.91 | 0.99 | 0.99 | 0.93 | 0.99 | 0.98 | 0.95 | 0.98 | 0.91 | 0.958 |
| No. Mild | 0.58 | 0.75 | 1.01 | 0.96 | 0.78 | 0.92 | 0.90 | 1.14 | 1.00 | 1.00 | 0.91 | 0.95 | 0.76 | 0.86 | 0.65 | 0.866 |
| No. Severe | 0.21 | 0.69 | 0.99 | 1.18 | 0.59 | 0.88 | 0.73 | 1.23 | 0.77 | 0.96 | 0.85 | 0.92 | 1.21 | 0.58 | 1.11 | 0.801 |

Table 5: The results of perimeter-degree constrained clustering formulation in terms of routed wirelength, place wirelength, the number of mildly congested edges and the number of severely congested edges are compared against the default placement and provided here as ratios. The last column provides the geometric mean of these ratios.

| | m01 | m02 | m03 | m04 | m05 | m06 | m07 | m08 | m09 | m10 | m11 | m12 | m13 | m14 | m15 | Geo.Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RouteWL | 0.95 | 0.97 | 1.00 | 0.98 | 0.93 | 1.00 | 0.91 | 0.99 | 0.99 | 0.98 | 1.02 | 0.98 | 0.98 | 0.99 | 0.94 | 0.973 |
| PlaceWL | 0.95 | 0.98 | 1.00 | 0.98 | 0.92 | 0.99 | 0.94 | 0.99 | 0.99 | 0.98 | 1.02 | 0.98 | 0.98 | 1.00 | 0.94 | 0.974 |
| No. Mild | 0.89 | 0.79 | 0.97 | 0.97 | 0.73 | 1.02 | 0.84 | 0.99 | 0.99 | 1.01 | 1.03 | 0.94 | 0.96 | 0.89 | 0.64 | 0.903 |
| No. Severe | 0.74 | 0.63 | 0.97 | 0.88 | 0.42 | 0.90 | 0.80 | 0.94 | 0.79 | 1.00 | 0.94 | 0.91 | 0.87 | 0.92 | 0.71 | 0.810 |

Table 6: The results of cluster filtering technique in terms of routed wirelength, place wirelength, the number of mildly congested edges and the number of severely congested edges are compared against the default placement and provided here as ratios. The last column provides the geometric mean of these ratios.

#### 5.2.2 Filtering Clusters

The impact of the idea of evaluating more pairings of clusters than needed and select a subset of them after discarding high perimeter-degree pairings (Section 4.2.2) is shown in Table 6. The RouteWL and PlaceWL have reduced by 3% on average, while the number of mild congested edges and the number of severe congested edges are reduced by 10% and 19% respectively. An important observation is none of the benchmarks showed degradation in the number of severe congested edges.

### 5.3 Empirical Evaluation of Area Re-distribution Based Technique

In Section 4.3, we presented a scheme that in the finer levels of the placement re-distributes the area of the cells such that the perimeter-degree value of all cells is uniform. In this section, we present the empirical validation of this effective scheme in Table 7. Very similar to the white space injection schemes, this scheme also does not show any change in wirelength metrics. The number of mild congested edges and severe congested edges are lower by 12% and 32% respectively. Notice the reduction of 32% in severe congested edges is the best average reduction in severe congestion among all the techniques presented.

| | m01 | m02 | m03 | m04 | m05 | m06 | m07 | m08 | m09 | m10 | m11 | m12 | m13 | m14 | m15 | Geo.Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RouteWL | 0.94 | 1.02 | 1.01 | 0.99 | 0.98 | 1.01 | 0.92 | 0.99 | 1.01 | 1.03 | 1.00 | 1.00 | 0.98 | 1.02 | 1.00 | 0.993 |
| PlaceWL | 0.93 | 1.02 | 1.00 | 0.99 | 0.99 | 1.01 | 1.00 | 1.00 | 1.04 | 1.03 | 1.00 | 1.00 | 0.99 | 1.02 | 1.00 | 1.001 |
| No. Mild | 0.65 | 1.02 | 0.96 | 0.77 | 0.81 | 0.99 | 0.86 | 0.79 | 0.99 | 1.02 | 0.80 | 0.97 | 0.73 | 1.09 | 0.94 | 0.884 |
| No. Severe | 0.34 | 1.14 | 0.90 | 0.39 | 0.78 | 0.84 | 0.97 | 0.67 | 0.87 | 1.05 | 0.58 | 0.85 | 0.26 | 0.86 | 0.59 | 0.685 |

Table 7: The results of area re-distribution scheme in terms of routed wirelength, place wirelength, the number of mildly congested edges and the number of severely congested edges are compared against the default placement and provided here as ratios. The last column provides the geometric mean of these ratios.

### 5.4 Overall Comparisons

Among the six techniques consisting of three white space aided variants, two modified clustering formulations and the area re-distribution schemes, the area re-distribution scheme obtained the best results. However, other schemes are also substantially effective in reducing congestion, and the white space aided techniques can further benefit from fine-tuning the threshold parameters.

Unlike the other techniques which showed practically no change in wirelength, the new clustering formulations enabled the refinement algorithm to find better wirelength. The reduction in wirelengh may also have reduced the congestion, in addition to the slightly less than stellar performance in congestion reduction compared to other techniques. Nevertheless, the new clustering formulation does not affect the utilization which makes it an attractive choice for highly utilized designs.

The overall runtime of the placement algorithm is dominated by the refinement algorithm due to its iterative nature coupled with the cost of evaluating actual half-perimeter cost for each move. The techniques presented in this paper added extremely insignificant overhead to the overall global placement runtime. Therefore, we do not provide any runtime comparisons.

## 6 Concluding Remarks

One of the sanity checks that need to be performed to ensure the techniques presented are indeed effective in reducing the congestion induced by local variations in interconnect density, is to check for the change in total placement wirelength. Lets say there are two techniques $A$ and $B$ that reduce congestion by equal amount, but technique $A$ does not result in any wirelength change, but the technque $B$ does reduce the wirelength significantly. In this case, technique

$A$ is superior as we know for sure that all the reduction in congestion is from influencing the interconnect density variations, in contrast, in the case of technique $B$, the answer to the question of whether the congestion reduction is due to reduced variations of interconnect density or due to overall reduction of total wirelength is far from certain.

Except in the case of new clustering formulations that reduced the wirelength by upto 4% on average, all other schemes showed practically no change in the wirelength. That clearly shows that the heuristics presented in this paper are extremely effective in reducing the variations in interconnect density.

# References


[1] DOLPHIN: A Complete Physical Implementation Tool Providing RTL to GDSII conversion in a single tool, www.mondes.com.

[2] C. J. Alpert, J. H. Huang, and A. B. Kahng. Multilevel circuit partitioning. In *Proc. of DAC*, 1997.

[3] U. Brenner and A. Rohe. An effective congestion driven placement framework. In *Proc. of DAC*, 2002.

[4] A. Caldwell, A. Kahng, and I.L.Markov. Can recursive bisection alone produce routable placements? In *Proc. of DAC*, pages 477–482, 2000.

[5] A. Caldwell, A. Kahng, and I. Markov. Relaxed partitioning balance constraints in top-down placement. In *Proc. of ASIC Conference*, pages 229–232, 1998.

[6] T. Chan, J. Cong, T. Kong, and J. Shinner. Multilevel optimization for large-scale circuit placement. In *Proc. of ICCAD*, pages 171–176, 2000.

[7] C. Chang, J. Cong, Z. Pan, and X. Yuan. Physical hierarchy generation with routing congestion control. In *Proc. of ISPD*, pages 36–41, 2002.

[8] J. Cong(Editor) and J. Shinnerl(Editor). Chapter3: Multilevel hypergraph partitioning. In *Multilevel Optimization in VLSI-CAD*, 2003.

[9] J. Davis, V. De, and J. Meindl. A stochastic wire-length distribution for gigascale integration (gsi) - part i: Derivation and validation. *Trans. on Electron Devices*, 45(3):580–589, 1998.

[10] W. Donath. Placement and average interconnection lengths of computer logic. *Trans. on Circuits and Systems*, CAS-26:271–277, 1979.

[11] H. Eisenmann and F. Johannes. Generic global placement and floor planning. In *Proc. of DAC*, pages 269–274, 1998.

[12] C. Fiducia and R. Matheyses. A linear time heuristic for improving network partitions. In *Proceedings of DAC*, pages 175–181, 1982.

[13] W. Hou, H. Yu, X. Hong, Y. Cai, W. Wu, J. Gu, and W. Kao. A new congestion-driven placement algorithm based on cell inflation. In *Proc. of ASPDAC*, pages 605–608, 2001.

[14] B. Hu and M. Marek-sadowska. Congestion minimization during placement without estimation. In *Proceedings of ICCAD*, pages 737–745, Nov 2002.

[15] P. Hung and M. Flynn. Stochastic congestion model for vlsi systems. Technical Report CSL-TR-97-737, Stanford University, 1997.

[16] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. Multilevel hypergraph partitioning: Application in vlsi domain. *IEEE Transactions on VLSI Systems*, 20(1), 1999. A short version appears in the proceedings of DAC 1997.

[17] G. Karypis and V. Kumar. Multilevel $k$-way hypergraph partitioning. In *Proceedings of the Design and Automation Conference*, 1999.

[18] B. Landman and R.Russo. On a pin versus block relationship for partition of logic graphs. In *IEEE Trans. on Computers*, pages C–20:1469, 1971.

[19] J. Lou, S. Krishnamoorthy, and H. Sheng. Estimating routing congestion using probabilistic analysis. In *Proc. of ISPD*, pages 112–117, 2001.

[20] H. Marck, D. Stroobandt, and V. Campenhout. Towards an extension of rent's rule for describing local variations in interconnection complexity. In *Proc. of 4th Intl Conf. for Young Scientists*, pages 136–141, 1995.

[21] P. Parakh, R. Brown, and K. Sakallah. Congestion driven quadratic placement. In *Proc. of DAC*, pages 275–278, 1998.

[22] L. Sanchis. Multiple-way network partitioning. *IEEE Trans. On Computers*, 38(1):62–81, 1989.

[23] N. Selvakkumaran, P. Parakh, and G. Karypis. Perimeter-degree: A priori metric for directly measuring and homogenizing interconnection complexity in multilevel placement. In *Proceedings of SLIP '03*, pages 53–59. ACM, April 2003.

[24] D. Stroobandt and J. Campenhout. Accurate interconnection length estimations for predictions early in the design cycle. *VLSI Design, Special Issue on Physical Design in Deep Submicron*, 10(1):1–20, 1999.

[25] P. Suaris and G. Kedem. Quadrisection: A new approach to standard cell layout. In *Proc. of ICCAD*, pages 474–477, 1987.

[26] J. Vygen. Algorithms for large-scale flat placement. In *Proc. of DAC*, pages 746–751, 1997.

[27] M. Wang, X. Yang, and M. Sarrafzadeh. On the behavior of congestion minimization during placement. In *Proc. of ISPD*, pages 145–150, 1999.

[28] M. Wang, X. Yang, and M. Sarrafzadeh. Dragon2000: Standard-cell placement tool for large industry circuits. In *Proc. of ICCAD*, pages 160–163, 2000.

[29] S. Wichlund and E. J. Aas. On Multilevel Circuit Partitioning. In *Proc. of ICCAD*, 1998.

[30] X. Yang, B. Choi, and M. Sarrafzadeh. Routability driven white space allocation for fixed-die standard-cell placememnt. In *Proc. of ISPD*, 2002.

[31] X. Yang, R. Kastner, and M. Sarrafzadeh. Congestion estimation during top-down placement.

[32] M. Yildiz and P. Madden. Global objectives for standard cell placement. In *Proc. of GLSVLSI*, pages 68–72, 2001.

[33] P. Zarkesh-Ha, J. Davis, and J. Meindl. Prediction of net-length distribution for global interconnects in a heterogeneous system-on-a-chip. *IEEE Trans. On VLSI*, 8(1):649–659, 2000.

[34] K. Zhong and S. Dutt. Algorithms for simultaneous satisfaction of multiple constraints and objective optimization in a placement flow with application to congestion control. In *Proc. of DAC*, pages 854–859, 2002.

[35] K. Zhong and S. Dutt. Effective partitioning-driven placement with simultaneous level processing and global net views. In *Proc. of ICCAD*, pages 254–259, 2002.