Collaborative Multi-Objective Global Routing

Hamid Shojaei, *Student Member, IEEE,* Azadeh Davoodi, *Member, IEEE,* Twan Basten, *Senior Member, IEEE*

***Abstract*— This work presents a collaborative procedure for multi-objective global routing. Our procedure takes as input multiple global routing solutions, which are generated independently (e.g., by one router that runs in different modes concurrently, or by different routers running in parallel). It then performs multi-objective optimization based on Pareto algebra and quickly generates multiple global routing solutions with a tradeoff between the considered objectives. The user can control the number of generated solutions and the degree of exploring the tradeoff between them by constraining the maximum allowable degradation in each objective. This work then considers the following three multi-objective case studies: minimization of interconnect power and wirelength, minimization of routing congestion and wirelength, and minimization of wirelength with respect to the (finite-capacity) routing resources. The maximum allowable degradation in wirelength is specified in all cases. Our multi-objective procedure runs in only a few minutes for each of the ISPD 2008 benchmarks, even for the unroutable ones, which imposes a tolerable overhead in the design flow. In our simulations, we demonstrate the effectiveness of our procedure using five modern academic global routers.**

***Index Terms*— Global routing, Interconnect power, Multi-objective, Pareto Algebra, Congestion spreading.**

I. Introduction

GLOBAL Routing is increasingly gaining importance to combat obstacles such as timing and congestion when realizing complex systems in the nanometer design era. Ever since the release of large industrial benchmarks in the ISPD 2007 contest [11], new global routers have emerged which have pushed the boundaries to obtain higher solution quality and faster runtime [2], [3], [4], [5], [6], [9], [10], [13], [14], [18], [20], [21], [27], [28], [30], [23], [24]. Most recently, the work [29] proposes a multi-level global router with significant improvements both in wirelength and runtime.

The global routing procedures can be divided into two categories of concurrent and sequential procedures. The concurrent procedures consider simultaneous routing of all the nets [2], [5], [18], [27], [28], while the sequential ones impose an ordering to route the nets [3], [4], [9], [10], [13], [20], [21], [30]. Furthermore, the sequential techniques apply drastically different procedures for different steps such as net ordering, layer assignment, and route generation.

These tools are inherently different in the underlying procedures to tackle the routing problem. Each one provides exclusive features which result in distinct routing solutions compared to the other one. For example, [16] points out that for benchmark circuit `adaptec1`, compared to the tools FGR and BoxRouter, the tool MaizeRouter [15] tends to route the nets above the blockages and close to the edges of the chip. However, despite distinct routing solutions, the measured qualities such as wirelength are not very different among the majority of recent global routing tools.

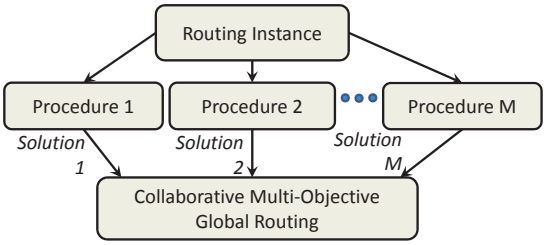

Fig. 1. Overview of our collaborative multi-objective framework.

Distinction in the solutions provides opportunity for generating a better one, for example when considering secondary objectives beyond wirelength. However, combining various routing procedures is difficult, if not impossible, since they may be incompatible (e.g., sequential versus concurrent).

In this work, we show that it is possible to only combine these different solutions, which are reflective of these procedures in order to conduct multi-objective optimization beyond the traditional wirelength optimization. As shown in Figure 1, different solutions can be obtained by running different routing procedures in parallel which could be from different tools or by concurrently running the tool from the same vendor in different modes. Examples of different routing modes include optimizing wirelength, or routing overflow, or congestion. Our collaborative procedure is able to combine these solutions to perform multi-objective optimization. Each routing solution has a 3D route for each net, which may span multiple metal layers. Our procedure directly works with these 3D input routes to generate a a new 3D routing solution which has varying degrees of contributions of its different input solutions. It effectively explores the search space and tradeoff between the objectives and outputs multiple global routing solutions which provide a tradeoff set between the objectives. Some degree of control in exploring the tradeoff space is also possible; an upper bound on each objective can be specified and the number of generated solutions can be controlled.

One feature of our procedure is its execution runtime which is consistently a few minutes for each of the ISPD 2008 benchmark circuits [12] including the unroutable benchmarks. Assuming the single-objective input procedures run independently in parallel, the additional runtime imposed by our pro-

An extended abstract of this paper was published in the 2010 International Symposium on Low Power Electronics and Design (ISLPED) [26].

The authors, Shojaei and Davoodi, are with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI 53706 USA (e-mail: shojaei@wisc.edu; adavoodi@wisc.edu), and the author, Twan Basten, is with the Department of Electrical Engineering, Eindhoven University of Technology & the Embedded Systems Institute, Eindhoven, The Netherlands (a.a.basten@tue.nl).

This work was in part supported by the National Science Foundation under Award 0914981.

Router	Net Decomposition	Two-terminal Routing	Net Ordering	Layer Assignment
NTHU-R 2.0 [4]	SMT (Steiner Minimal Tree)	Pattern, Maze Routing	Current Congestion	COLA [14]
BFGR [10]	MST (Minimal Spanning Tree)	Maze Routing	Bounding Box, Current Congestion	Greedy
FastRoute 4.0 [28]	SMT	Pattern, Maze Routing	Bounding Box	DP (Dynamic Programming)
MGR [27]	SMT	Pattern, 3D & 2D Maze	Bounding Box	DP, 3D Routing
NCTU-GR [6]	SMT	Monotonic, Maze Routing	CFOMR (Circular Fixed Ordering Monotonic Routing)	CRLA (Congestion-Relaxed Layer Assignment)

Fig. 2. Comparison of collaborating global routers used in this work.

cedure is tolerable in the design flow. Our procedure provides multi-objective capability using single-objective procedures.

Specifically, this work identifies practical variations of multi-objective global routing. When using five academic global routers to generate the initial routing solutions, we show significant and consistent improvement for different multi-objective cases. (The comparison between the collaborating routers used in our simulation results is given in Figure 2.)

In one case we explore simultaneous optimization of wirelength and interconnect power. The interconnect power is measured using a metric which is based on utilization of each edge in the routing grid-graph while considering wire spacing at each metal layer for modeling the coupling capacitance contribution to power. The user can explore the tradeoff between increase in wirelength and decrease in power. This is subject to a specified upper bound on the allowable wirelength degradation. In our experiments, when wirelength degradation is not allowed by the user, we show that on average 9.72% improvement in the power metric is achievable while wirelength is also improved by 0.29%. Our framework outputs multiple routing solutions and we illustrate the tradeoff between wirelength and power in up to four stored solutions for each benchmark. The runtime of a multi-objective power-wirelength experiment is on average 4.65 minutes on the ISPD 2008 benchmark circuits [12].

We also consider simultaneous optimization of wirelength and routing congestion—measured as the total utilization of congested edges in the routing grid-graph. In our experiments conducted on the ISPD 2008 benchmark instances, we show that the utilization on the congested edges is reduced, on average by 7.13% without any degradation in wirelength and even the wirelength is improved by on average 0.26%.

A third variant of our framework is used for multi-objective optimization of wirelength and routing resources. In this variant, the amount of total wirelength is not considered to be equivalent to routing resource usage. This can be due to different factors such as distinction in routing resource usage of the metal layers and inter-layer vias.

In summary, this work offers a collaborative procedure which combines input routing solutions for multi-objective optimization. It generates multiple solutions which reflect a tradeoff between the considered objectives.

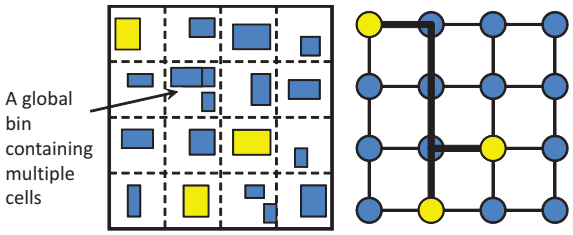

Fig. 3. Grid-graph model of global routing.

The remainder of the paper is organized into the following sections. Section II describes the problem formulation. Section III describes our collaborative procedure to solve the formulation. Section IV discusses several variations of the framework. Simulation results are presented in Section V and a summary of concluding remarks are offered in Section VI.

II. Problem Formulation

We first discuss a mathematical description of the (single-objective) Collaborative Global Routing problem and explain the multi-objective solution procedure in the subsequent section. We are given a routing grid graph $G = (\mathcal{V}, \mathcal{E})$. As shown in Figure 3, a node represents a global bin containing multiple cells. An edge represents the boundary between two adjacent global bins. Each edge e has a capacity r_e, representing the number of routes that can pass from the boundary of the two adjacent global bins. A set of N nets given by $\mathcal{N} = \{n_1, n_2, \ldots, n_N\}$, (with $n_i \subseteq \mathcal{V}$), are also specified. The graph G has 3 dimensions to capture the multi-layer routing.

We use $\mathcal{T}_i$ to denote a set representing the collection of candidate routes for net n_i. These candidate routes are provided a-priori based on the flow given in Figure 1. Each routing procedure generates a 3D route for each net which may span multiple metal layers.

We denote by $\mathcal{T}_i(k)$ the routing solution of procedure k for net n_i so we have $\mathcal{T}_i(k) \in \mathcal{T}_i,\ \forall k = 1, \ldots, M$, where M is the number of routing procedures. It may be possible that two procedures generate the same route for a net in their solutions (e.g., $\mathcal{T}_i(k) = \mathcal{T}_i(j)$ indicates that the same route is obtained for net n_i by the tools k and j). In this case, the set $\mathcal{T}_i$ will include this common route only once. Therefore, the number of candidate routes for n_i can be at most M: $|\mathcal{T}_i| \leq M$.

Each candidate route is a 3D Steiner tree connecting the terminals in n_i. For an arbitrary Steiner tree t, let $a_{te} = 1$ if it contains edge $e \in \mathcal{E}$; $a_{te} = 0$ otherwise. Define the binary variable x_{it} to be equal to 1 if and only if $t \in \mathcal{T}_i$ is selected for net n_i. The Collaborative Global Routing (CGR) problem is written as the following Integer Linear Program (ILP):

$$\min_{x} \sum_{i=1}^{N} \sum_{t \in \mathcal{T}_i} p_{it} x_{it} \tag{1}$$

$$\begin{cases} \sum_{t \in \mathcal{T}_i} x_{it} = 1 & \forall i = 1, \ldots, N \\ \sum_{i=1}^{N} \sum_{t \in \mathcal{T}_i} a_{te} x_{it} \leq r_e + o_e & \forall e \in E \\ \sum_{i=1}^{N} \sum_{t \in \mathcal{T}_i} wl_{it} x_{it} \leq WL_{th} & \\ x_{it} \in \{0, 1\} & \forall i = 1, \ldots, N, \forall t \in \mathcal{T}_i. \end{cases}$$

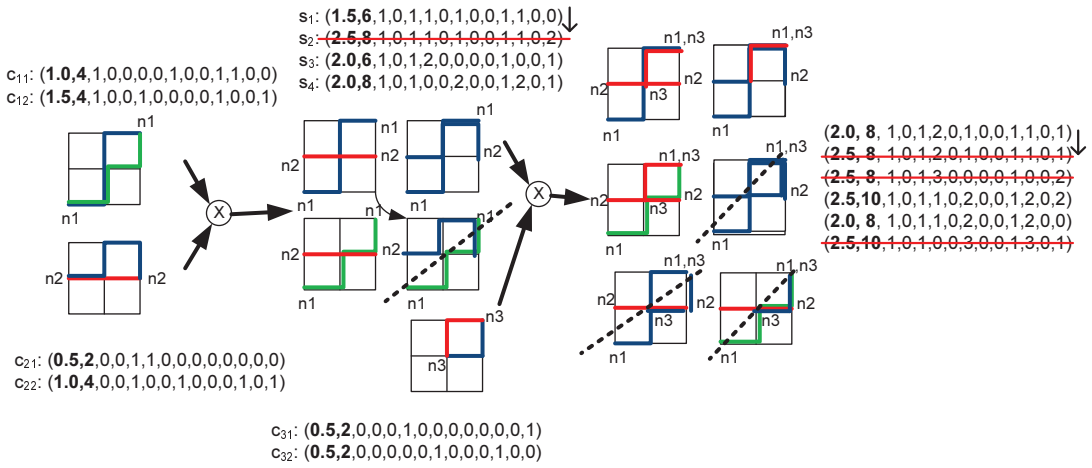

Fig. 4. Example of our Pareto-algebraic procedure for the CGR problem.

where p_{it} is a constant parameter representing the cost (penalty) of route t for net n_i. The objective is to minimize the summation of the costs of the routed nets. For example, if p_{it} is the length of Steiner tree t, then the objective is to minimize the total wirelength. Cost could be alternatively defined, e.g., associated with an edge (instead of a net). For example, it can be the number of heavily-utilized routing edges, or the overall power. We elaborate when we discuss variations of the CGR formulation in Section IV.

In the above formulation, the first set of equations enforces selection of one candidate route for each net. In the second set of equations, o_e is a constant parameter. It represents the overflow of edge e corresponding to the routing solution of one of the M procedures. If procedure k is the one considered as reference, then o_e is given by

$$o_e = \max(0, \sum_{i=1, t=\mathcal{T}_i(k)}^{N} a_{te} - r_e). \tag{2}$$

In this work we assume the selected procedure is the one which results in the smallest total overflow routing solution. So for example, in the case when one of the procedures results in a zero overflow solution for all the nets, then all the o_e parameters will be set to zero. Based on our definition of o_e, the quantity $r_e + o_e$ can be thought of as an adjusted (and possibly increased) edge capacity. The second set of equations thus ensures that the selected candidate routes of all the nets do not exceed this adjusted edge capacities. This definition of edge capacity guarantees that a feasible solution always exists for the CGR ILP formulation.

In the third set of equations, the parameter wl_{it} represents the wirelength of route t for net n_i. The third equation bounds the total wirelength of the selected routes by a user-defined input WL_{th}. Thereby, minimization of the objective function can be done with respect to a *tolerable degradation factor* compared to an initial wirelength, for example corresponding to one of the routing tools. This is because wirelength is the prominent objective of global routing. Optimizing other objectives should be done by controlling the impact on wirelength.

The formulation of CGR is in the form of the multi-dimensional multiple-choice knapsack problem (MMKP) [17]. The MMKP is in the class of NP-complete problems. In [25], a Pareto-algebraic heuristic is given to quickly solve MMKP instances with high quality. Pareto algebra [7] allows to reason about multi-objective optimization problems in an algebraic manner. This translates to a compositional, incremental solution approach to MMKP. In this work, we extend the procedure in [25] and adapt it for the CGR problem. Furthermore, our procedure is applicable to the case when multiple objectives are simultaneously optimized. It generates multiple global routing solutions and a corresponding tradeoff set between the objectives. We elaborate the multi-objective case as we discuss our procedure in the next section.

III. Collaborative Global Routing Procedure

This section presents our collaborative global routing procedure. We first describe an encoding of candidate routes as *configurations* of a net to apply our procedure as an MMKP instance and then present the details of the procedure.

A. Modeling and Overview of Our Procedure

In a Pareto-algebraic description of CGR, each candidate route of a net is represented in terms of a configuration of $P + R$ dimensions denoted by $(d_1, d_2, \ldots, d_{P+R})$. For P considered cost (penalty) functions, the first P dimensions represent the corresponding costs of that configuration. For example for $P = 2$, the first and second dimensions can be the wirelength and power of the candidate route, respectively. For the global routing problem, most of the dimensions of a configuration are equal to zero so a sparse representation is used to store a non-zero dimension as a pair of the index and corresponding value.

Furthermore, R is the number of edges in the routing grid-graph, excluding those corresponding to inter-layer vias. If a candidate route is passing an edge, then the corresponding dimension will be a 1 entry in its configuration, and otherwise it is 0. Therefore the summation of the last R entries in

the configuration reflects the routing resource usage of the candidate route. It could be smaller than its wirelength if the vias are assumed to have infinite capacity, similar to the ISPD global routing contest [11] and the majority of subsequent academic routing procedures are based on this setup.

We denote the j^{th} configuration of net n_i by $c_{i,j}$. For example, in Figure 4 (left), we have 2 candidate routes for net n_1 and 2 for net n_2. Each route has therefore two configurations. The locations of the net terminals are marked on the 2x2 routing grid-graph. We assume $P = 2$ for two objectives of power and wirelength which are given by the first two dimensions. The remaining $R = 12$ entries reflect the utilization of the edges in the grid-graph; for a candidate route, there is an entry of 1 in a dimension if it contains the corresponding edge.

The set of candidate routes for a net n_i thus composes its *configuration set* which we denote by $\mathcal{C}_i = \{c_{i,1}, ..., c_{i,|\mathcal{C}_i|}\}$. For the CGR problem, we assume $|\mathcal{C}_i| \leq M; \forall i = 1, \dots N$, where M is the number of input routing procedures. Configuration $c_{i,j}$ is a Pareto point of set $\mathcal{C}_i$ if it is not dominated by any other configuration point. A configuration point dominates another point iff it is at least as good as that point in all dimensions and better in at least one dimension. Pareto algebra defines operations on configuration sets in order to compute Pareto points. For more details, we refer the reader to [8].

We proceed to an overview of our procedure to solve the multi-objective CGR formulation as an MMKP instance. We traverse the nets once in a specific order (discussed in Section III-C). At each step of computation, we combine the configurations of the current net with an existing configuration set corresponding to all the nets processed in the previous steps. During each step we maintain a set of options for all nets processed up-to that point, each option providing a different tradeoff in the P objectives and the R resources. After each step, we thus obtain a set of compound configurations denoted by $\mathcal{S} = \{s_1, s_2, ..., s_{|\mathcal{S}|}\}$ where s_i is one compound configuration. Conceptually, each combination of two configuration sets corresponds to a product of these sets.

Each compound configuration is also represented by a tuple of $P+R$ dimensions. As before, the first P dimensions give the corresponding costs of the compound configuration after the combination. The last R dimensions give the resource usage (or utilization) of each intra-layer edge in the routing grid-graph after the combination; they are obtained by element-wise addition of the corresponding dimensions in the combined configurations. For example, when combining two configurations s_i and s_j, the last R entries in the compound configuration are simply the element-wise addition of the last R entries in s_i and s_j, indicating that the utilization of each edge will be the summation of its utilizations in s_i and s_j. Similarly, the wirelength is always the summation of the wirelength dimensions in s_i and s_j. However, the P cost dimensions of the compound configuration may be a function of all the other dimensions of s_i and s_j in the general case. We discuss variations of CGR for different cost functions and elaborate computation of the compound configurations in Section IV.

After obtaining a compound configuration, a feasibility check is made and the configuration will be removed if it is infeasible. Specifically, if the wirelength (first dimension) is larger than the specified upperbound (WL_{th}) or if the utilization of an edge (any of the last R entries) is larger than its adjusted capacity ($o_e + r_e$), then the compound configuration is infeasible and will be removed. Furthermore, configurations that are dominated in terms of all the dimensions are also removed, i.e., only Pareto points are kept.

Consider Figure 4 again which shows a snapshot of the procedure for a small example of nets n_1, n_2, n_3 on the 2x2 grid-graph. Each edge has an adjusted capacity of 2 units. First, net n_1 and net n_2 (each with 2 configurations) are combined. Among the 4 compound configurations, s_2 is dominated by s_1 and is thus removed. Net n_3 with 2 configurations is then combined with the configuration set (of size 3) representing nets n_1 and n_2. Next, we obtain 6 configurations. The last step then removes 3 of these configurations (where 1 configuration is removed because it is dominated and 2 other configurations are infeasible due to edge capacity violation).

B. Details of the Procedure

Algorithm 1 gives the details of the described procedure. Specifically, we consider the MMKP procedure given in [25] for runtime management in chip-multiprocessors, and introduce new considerations to solve the CGR problem.

For each net n_i in $\mathcal{N}$, with $1 \leq i \leq N$, we require an initial configuration set $\mathcal{C}_i = \{c_{i,1}, \dots, c_{i,|\mathcal{C}_i|}\}$, where the configurations correspond to the routes from the M routing procedures. In addition, we take as input the adjusted edge capacity vector denoted by $r_{\mathcal{E}}$, and two user-defined parameters L and WL_{th}. The former controls the execution runtime of the algorithm (as explained in Section III-D) while the latter defines the wirelength threshold.

The set of compound configurations $\mathcal{S}$ is initially set to $\mathcal{C}_1$ for net n_1 (line 2). This means that element $s_j = c_{1,j}, \forall s_j \in \mathcal{S}$. The remaining nets are then traversed once. We discuss the traversal ordering in Section III-C. For each net n_i (line 3), we consider each $s_j \in \mathcal{S}$, reflecting the combinations of processed nets n_1 to n_{i-1} with configurations of n_i (lines 5, 6).

First we check if the compound configuration corresponding to the combination of $c_{i,l}$ and s_j is feasible (line 7). To check for feasibility, we first compute the last R dimensions by element-wise additions of the corresponding dimensions in $c_{i,l}$ and s_j and check for satisfaction of the edge capacities.

We also compute the wirelength of the compound configuration by adding the wirelength dimensions in $c_{i,l}$ and s_j and compare with WL_{th}. If the feasibility checks are successful, we proceed to combine $c_{i,l}$ and s_j which is denoted by the compound configuration s^{tmp}. To obtain s^{tmp} (line 8) we need to compute the remaining dimensions which correspond to the values of the cost functions in the configurations considered for nets n_1 to n_{i-1} (given in s_j) and for net n_i (given in $c_{i,l}$).

In Section IV we discuss the implementation of $combine$ (line 8) for different cost functions, corresponding to different variations of CGR. After the combination, the result is added to the set $\mathcal{S}^{tmp}$, reflecting the set of feasible combinations between the configurations of n_i and the s_js processed so far. The Pareto-algebraic min operation is applied to $\mathcal{S}^{tmp}$ to

Algorithm 1 Pareto-algebraic heuristic for CGR

1: CGR($\mathcal{C}_i$ $(\forall n_i \in \mathcal{N})$, $r_\mathcal{E}$, L, WL_{th})
2: $\mathcal{S} = \mathcal{C}_1$ *//so we have* $(s_1 = c_{1,1}, s_2 = c_{1,2}, \ldots, s_{|\mathcal{C}_1|} = c_{1,|\mathcal{C}_1|})$
3: **for all** net n_i, $i = 2$ to $|\mathcal{N}|$ *// Sec. III-C* **do**
4: $\quad\mathcal{S}^{tmp} = \emptyset$
5: $\quad$**for all** $l = 1$ to $|\mathcal{C}_i|$ **do**
6: $\quad\quad$**for all** $s_j \in \mathcal{S}$ **do**
7: $\quad\quad\quad$**if** feasible($s_j, c_{i,l}, r_\mathcal{E}, WL_{th}$) **then**
8: $\quad\quad\quad\quad s^{tmp}$ = *combine*($s_j, c_{i,l}$) *// Sec. IV*
9: $\quad\quad\quad\quad\mathcal{S}^{tmp} = \mathcal{S}^{tmp} \cup \{s^{tmp}\}$
10: $\quad\quad\quad\quad min(\mathcal{S}^{tmp})$ *// Sec. IV*
11: $\quad\quad\quad$**end if**
12: $\quad\quad$**end for**
13: $\quad$**end for**
14: $\quad\mathcal{S} = \mathcal{S}^{tmp}$
15: $\quad reduce(\mathcal{S}, L)$ *// Sec. III-D*
16: **end for**

get the Pareto points. This boils down to checking whether the newly-added configuration is dominated or whether it dominates any of the already-present configurations. Since the number of dimensions $(P + R)$ could be very large for the CGR problem, applying the min operation considering all the dimensions is highly time-consuming. We therefore approximate min by applying it in a 2-dimensional space. We discuss this in Section IV for different CGR variations.

The process repeats to combine the next configuration s_{j+1} with n_i. When all s_js are combined with the configurations of n_i, set $\mathcal{S}$ is updated to $\mathcal{S}^{tmp}$ and net n_{i+1} is processed.

For implementation efficiency, we apply the min operation after combining each feasible s_j and $c_{i,l}$ (instead of a one-time min after all the configurations are combined). This allows reduction in the number of intermediate non-Pareto configurations, which makes each iteration of the loops faster. This implementation of minimization is following the *Simple Cull* approach [31]. As shown in [7], Simple Cull is in practice the most efficient implementation.

However, the above procedure is still intractable in many cases, as the size of $\mathcal{S}$ can become very large. To speed up the computation, we check if the size of $\mathcal{S}$ becomes larger than the user-defined threshold parameter L. We only keep at-most L configurations and prune the rest (line 15). We explain this step in Section III-D. In the end, a tradeoff set of the considered objectives is provided to the user. The set has at-most L points. Our implementation allows each point to be tracked back to generate the corresponding configuration of each net, and thus a complete global routing solution. The user can pick a desired point and the algorithm generates the final solution.

C. Net Ordering

Algorithm 1 relies on an ordering for processing the nets. Based on our experience, we observe that the ordering can highly impact the quality of the final solution. Selecting a suitable ordering may not be trivial since the single-objective routing procedures may use different orderings which may be incompatible with each other. We observe that for our Pareto-algebraic procedure, a good ordering is one which can identify and prune the infeasible and dominated solutions early in the combination steps. Otherwise, the number of Pareto points will quickly reach the threshold and the reduction step will need to be applied to remove the Pareto points.

Algorithm 2 Net ordering procedure

1: NETORDERING($\mathcal{C}_i(\forall n_i \in \mathcal{N})$, $\mathcal{N}$, $\mathcal{E}, k$)
2: **for all** $e \in \mathcal{E}$ **do**
3: $\quad u_e^{approx} = \sum_{i=1, t=\mathcal{T}_i(k)}^{N} a_{te}$
4: **end for**
5: $\mathcal{N}_{order} = \emptyset$; $\mathcal{E}_{candid} = \mathcal{E}$
6: **while** $|\mathcal{N}_{order}| \neq |\mathcal{N}|$ **do**
7: $\quad e_{crit}$=arg max$_{e \in \mathcal{E}_{candid}}(u_e^{approx})$
8: $\quad \mathcal{N}_{crit}$= $\{n_i \in \mathcal{N} |$ e_{crit} included in $c_{i,j}$, $1 \leq j \leq |C_i|\}$.
9: $\quad \mathcal{N}_{order} = \mathcal{N}_{order} \circ$ sort-decreasing-wirelength($\mathcal{N}_{crit}$)
10: $\quad \mathcal{E}_{candid} = \mathcal{E}_{candid} - \{e_{crit}\}$
11: **end while**

A good ordering can be achieved by looking at the congested areas first which can be *approximated* from the solutions of the routing procedures. In a congested area, the routing grid edges are utilized to close to their capacities, and therefore it will be more likely to detect and remove infeasible and dominated configurations in the first few iterations of the main loop of Algorithm 1. In addition, the congested areas include many more nets and may contribute more to the considered cost functions such as overall wirelength or power.

Algorithm 2 gives the pseudo-code of our ordering procedure. It receives the configuration set of each net and the sets of nets and edges as input. It also receives as a parameter a routing procedure k used as reference. As shown in lines 2 to 4 of Algorithm 2, for each edge, an approximate utilization is computed as the utilization of that edge in the solution of the routing procedure k. This approximate utilization is then used to drive the net ordering.

During ordering, we keep updating an ordered subset of nets $\mathcal{N}_{order} \subseteq \mathcal{N}$ until $|\mathcal{N}_{order}| = |\mathcal{N}|$ (line 6). We first identify a critical edge $e_{crit} \in \mathcal{E}_{candid}$ with highest approximate utilization u_e^{approx}, where initially $\mathcal{E}_{candid} = \mathcal{E}$ (line 5). Next, we form the subset $\mathcal{N}_{crit} \subseteq \mathcal{N}$ of critical nets where $n_i \in \mathcal{N}_{crit}$ if any of its configurations $c_{i,j}$ includes e_{crit} (line 8). Since the configurations of these routes include e_{crit}, they are more likely to result in pruning. We then sort the subset $\mathcal{N}_{crit}$ according to decreasing order of the wirelength of the nets, with each n in N_{crit} wirelength approximated by averaging the wirelengths corresponding to all its configurations $c_{i,j}$ (line 9). The sorted subset is then concatenated to the end of the existing $\mathcal{N}_{order}$ (also in line 9 where concatenation is denoted by the $\circ$ symbol). Finally, we update the set $\mathcal{E}_{candid}$ to find the next e_{crit} while excluding the previous e_{crit} identified from the previous iteration.

Using the above-described approximate utilization significantly improves our ordering procedure compared to other methods to estimate the utilization such as by looking at the overlap in the net bounding boxes. This is because of two reasons. First, the approximate utilization is inline with the adjustments applied to the edge capacities in the CGR formulation. Second, it corresponds to the utilization of one of the input routing solutions. In contrast, estimating the edge utilization in the existing sequential routing procedures is a

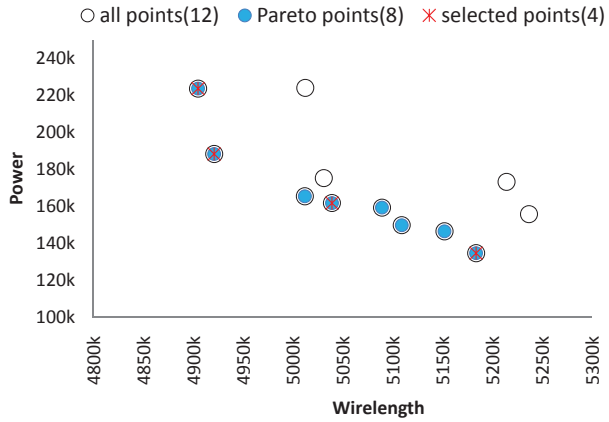

Fig. 5. Reduce after processing 150K nets in `adaptec1`.

more challenging task because they build a routing solution completely from scratch.

D. The Reduce Procedure

According to Algorithm 1, the set $\mathcal{S}$ representing the compound configurations as set in line 14 contains all the Pareto points. If the size of $\mathcal{S}$ exceeds the user-defined parameter L, we apply a procedure to drop some of these Pareto points. Our goal is to select L Pareto points (hence L compound configurations) such that they provide a tradeoff set between the considered cost functions and resources. Since the number of cost functions and resources $(P + R)$ is very large, in this work, we only discuss the reduce procedure for specific variations of CGR (discussed in Section IV). In all these variations, we project the compound configurations into a two-dimensional space. We will discuss the definition of each dimension for each variation separately (in Section IV). For example, when we consider two cost functions in the multi-objective CGR, each dimension represents one of the cost functions and each configuration is represented by a point in this space. We denote dimensions 1 and 2 by d_1 and d_2.

Figure 5 shows a snapshot of the projected compound set $\mathcal{S}$ after processing 150K nets in benchmark `adaptec1` for two cost functions of power and wirelength. To select L configurations, we first select two configurations, one with minimum d_1 value and the other with minimum d_2 value. We then select $L-2$ additional configurations as follows. For each configuration we determine a weight given by $\frac{d_1}{d_2}$ which characterizes the tradeoff between the two dimensions. In the above example, for configuration i, we define the weight $w_i = \frac{power_i}{wirelength_i}$.

We then try to cover the range of weights uniformly. We first select $L-2$ "reference values" according to the following formula: $\min_i(w_i) + \frac{\max_i(w_i) - \min_i(w_i)}{L-1} \times k,\ \forall k = 1, \ldots, L-2$. To select the remaining $L-2$ configurations, we visit these $L-2$ reference values one at a time. For reference value k, we select the configuration with a corresponding weight closest to this reference value. We remove the selected configuration and continue until all the reference values are considered. Therefore for $L-2$ reference values, we select $L-2$ configurations with weights which are closest to them. Together with the two initially-selected configurations, we obtain L configurations.

Figure 5 shows the selection of $L = 4$ in the above example. The procedure attempts to quickly and fairly approximate the case when none of the Pareto points are dropped.

E. Runtime Complexity

For M collaborating routers with up to L stored solutions and N nets, the runtime complexity of the CGR procedure is $O(N(ML \times M^2L^2 + M^2L^2log(ML))) = O(NM^3L^3)$. This is because for each of the N iterations of the algorithm (excluding the first one which can be ignored), the combination of L compound configurations representing the processed nets is considered with M configurations of the currently-considered net. For each combination, a Pareto minimization (line 10) is applied which is of $O(M^2L^2)$ for an input size of $O(ML)$. Then the *reduce* procedure requires a sorting and a walk through the compound configuration S (line 15) to pick L configurations. Therefore, given that the size of any of the compound configuration sets S is bounded by M^2L^2, the complexity of the $reduce$ step is $O(M^2L^2log(ML))$. Without the $reduce$ phase and Pareto minimization, the size of the compound configuration set exponentially grows and is of $O(M^i)$ for iteration i.

In practice, the runtime of the algorithm never shows its worst-case behavior. The Pareto minimization maintains a list of Pareto points observed *so far* every time one of the M configurations of the current net is combined with the current compound configuration set (line 10). As a result, the size of S^{tmp} is significantly reduced compared to its upper bound.

IV. Variations of Collaborative GR

In this section we study a few multi-objective variations of CGR. For each variation, we discuss the specific implementations of $combine$, min and $reduce$ in Algorithm 1. In Section V, we present the simulation results for each case.

A. Power and Wirelength Minimization

Power of signal interconnects (which excludes power-delivery and clocking networks) can have a significant contribution to the total power consumption in the nanometer era; for example, the contribution of the interconnect power is reported to be around 30% of dynamic power for a 45nm high performance microprocessor synthesized using a Structured Data Paths design style and about 18% of the overall power spectrum [22]. To model the signal power accurately, area and fringe capacitances for each wire segment should be considered depending on its width and metal layer. Furthermore, the spacing between the wires can determine their coupling capacitance which highly deteriorates with technology scaling. The above factors (i.e. metal layer and wire width) can be approximated during the global routing stage fairly accurately. Moreover, wire spacing can also be approximated at this stage by *approximating* the utilization of each edge within the global routing procedure. Therefore, opportunities lie in signal power optimization at this stage. Here, we consider multi-objective optimization of an interconnect power metric and wirelength.

Specifically, we first adjust the capacities of each edge in the global routing grid-graph to account for the wire size and spacing of its corresponding metal layer, as provided by the target technology. Typically for each layer a range of wire size options are available. In this work we assume the minimum wire size is used for each layer. This is because we use routing as the technique used for power minimization and do not consider wire sizing. After this adjustment, we consider the routes provided by the M routing procedures to be the candidate routes. We then apply CGR to optimize an interconnect power metric and wirelength.

The consideration of wire size and spacing reduces the edge capacities at the higher metal layers since these quantities are typically much larger at the higher layers. To compute the number of routes that pass the boundary of two adjacent global bins in a metal layer, we divide the length of the boundary of the two global bins by the summation of wire size and spacing of that layer. This gives a more accurate estimate of the number of routes that can pass the boundary which will be the adjusted edge capacity of the global routing edge corresponding to that boundary. Note, all the edges in the same metal layer will have the same capacity using this approach. We first review a power model for global routing (proposed by us in [26]).

1) Interconnect Power Model for Global Routing: The total interconnect power denoted by W can be expressed as the summation of dynamic powers of the routed nets:

$$W = V_{DD}^2 \times f_{clk} \times \sum_{i=1}^{|\mathcal{N}|} \alpha_i (C_i^{sink} + C_i^{wire}) \tag{3}$$

where V_{DD} is the supply voltage, f_{clk} is the frequency, and α_i is the switching activity of net i. The capacitance of a routed net i is the summation of the capacitances of its sink cells (denoted by C_i^{sink}), and of its wire (denoted by C_i^{wire}). The C_i^{wire} is expressed as the summation of all its unit wire segment capacitances and given by the equation below:

$$C_i^{wire} = \sum_{\forall j, e_j \in t_i} C_j^{unit} \tag{4}$$

where C_j^{unit} is the capacitance corresponding to edge e_j which is contained in route t_i. For a wire segment corresponding to edge e_j, we assume a width w_j and metal layer l_j. In the general case, the wire can be in the middle of two other neighboring segments running in parallel with spacing s_j, all of which are mapped to edge e_j, as shown in Figure 6. The capacitance corresponding to this wire segment is denoted by C_j^{unit} and is given by the equation below:

$$C_j^{unit} = C_a(l_j, w_j) + 2C_f(l_j, w_j, s_j) + 2C_c(l_j, w_j, s_j) \tag{5}$$

where C_a and C_f are the area and fringe capacitances with respect to substrate, and C_c is the cross-coupling capacitance between the wire segment and its parallel-running wires. Figure 6 illustrates these three types of capacitances.

The area, fringe and coupling capacitances are a function of metal layer l_j and wire width w_j corresponding to edge e_j. In addition, the fringe and coupling capacitances also depend on spacing s_j on edge e_j. For a given metal layer, wire width and spacing, the corresponding unit-length capacitance can then be

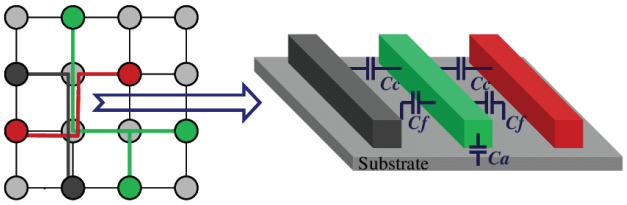

Fig. 6. Mapping edge utilization to interconnect capacitance.

computed typically using a lookup table provided by the technology library. For example, for the 45nm technology model used in this work [19], we observed that with decrease in spacing, C_c significantly increases while C_f slightly decreases. Also, with increase in wire width, C_a drastically increases. The wire width is also known from the library for each metal layer and a higher metal layer requires a higher wire width. The spacing depends on the routing congestion. Specifically, to compute the spacing at edge e_j at the global routing stage, we use the following formula:

$$s_j = \max(s_{min_j}, \frac{length_j}{\#wires_j} - w_j) \tag{6}$$

where $length_j$ denotes the length of the boundary between the two adjacent global bins that edge e_j represents, $\#wires_j$ denotes the number of wires that pass from e_j, w_j denotes the corresponding wire size, and s_{min_j} represents the minimum wire spacing for the layer that has e_j which is obtained from the lookup library. According to the above formula, we determine the spacing s_j such that the maximum distance between the wires that pass from e_j is obtained. This strategy minimizes the fringe and cross-coupling capacitances which makes sense if no spacing constraints are provided. On the other hand, if additional spacing constraints are provided, they can be incorporated for a more accurate estimate of wire spacing. For example, the adjustment may be a reduced capacity due to the presence of a fixed number of short (local) nets which completely fall inside a global bin.

Note, in the presence of overflow when edge e_j is utilized higher than its capacity, the spacing will be set to the minimum spacing s_{min_j} which is he smallest spacing that can be used for the edge according to the library model.

In summary, we estimate the spacing in the boundary of two adjacent global bins from the utilization of the corresponding edge in the global routing grid-graph. (During the global routing procedure, this utilization is further estimated as the nets are sequentially routed.) Together with the wire size and metal layer for that edge, we then look up the capacitance components from the library and use the presented equations to estimate the total signal power over all the nets. Therefore, to compute the power metric during global routing, the only requirement is the knowledge of the utilization of each edge in the grid-graph (given that the wire size and spacing are provided at each metal layer).

2) Details of the CGR Procedure: We assume the configuration of each net is represented by $2 + R$ dimensions. The first dimension is an *estimate* of total power. In this work,

we assume that each configuration of each net has for its first dimension as a value the total power estimated using the edge utilizations obtained from the solution of the reference routing procedure which was also used during net-ordering in Algorithm 2. The second dimension is the corresponding wirelength which is the length of the candidate route of that net. Similarly, in the compound configuration the first dimension is a more accurate estimate of total power (where the increased accuracy comes from taking into account the actually selected routes for the nets processed so far) and the wirelength is the length of the corresponding candidate routes represented by that compound configuration.

In Algorithm 1, to apply the $combine$ function to configuration $c_{i,l}$ and compound configuration s_j, we first update the last R fields by element-wise addition. We also compute the corresponding wirelength by adding the wirelength fields of $c_{i,l}$ and s_j. To compute the power field, we first obtain an *estimate* of the utilization of each edge by adding the utilization from the first i nets processed so far, with an estimate of the utilization for the remaining $N - i$ unprocessed nets. More specifically, for each edge e, we consider a utilization contributed by the processed nets n_1 to n_i, obtained by adding the corresponding fields in $c_{i,l}$ and s_j that represent edge e. We then estimate the remaining utilization of e contributed by nets n_{i+1} to n_N as estimated using configurations $c_{j(>i),k}$ (i.e., provided by the reference routing tool k). Once the utilization of each individual edge in the grid-graph is estimated, the power is computed as discussed before. Please note in this multi-objective variation, an estimate of the route corresponding to each unprocessed net is necessary in order to compute the utilization for each edge e_j and consequently its wire spacing according to Equation (6) for power estimation.

After the combination, to apply the min operation, each compound configuration in s^{tmp} is projected to a two-dimensional point with wirelength and power as the two dimensions. The Pareto minimization operation is then applied. Similarly, during the $reduce$ procedure, we assume d_1 to be wirelength and d_2 to be power consumption.

B. Congestion and Wirelength Minimization

We define a variation of CGR to minimize the utilization on those edges in the global routing grid-graph that are utilized close to or higher than their capacities. An edge is highly utilized if its utilization is higher than 70% of its capacity. We refer to such an edge as a HUT edge (i.e., highly-utilized). The goal is to minimize the summation of the utilization of the HUT edges simultaneously with wirelength. These two cost functions can be traded off with each other. According to [1], the HUT edges are likely to cause routability problems later during detailed routing. Minimizing the utilization of the HUT edges results in spreading the routing congestion and in turn increasing wirelength. The user is also allowed to define WL_{th} to control the maximum increase in wirelength.

In this variation, the configuration of each net has $2 + R$ dimensions where the first dimension is the total utilization of the HUT edges, and the second dimension is wirelength. Initially, for each net, the first field is 0 because the capacity of each edge is typically a large value and the rest of the nets are not considered. However, when considering the compound configurations, it is possible that some of the edges will become highly utilized. So the first field will likely have a non-zero entry in the subsequent steps of the algorithm. The second field represents the wirelength of a compound configuration.

In the $combine$ function, we first compute the last R fields of the combination of s_j and $c_{i,l}$ by element-wise addition. To compute the utilization of the HUT edges, we visit the last R fields after the combination. Recall each field represents the utilization in one of the intra-layer edges which can be compared against the capacity of that edge to determine if it is a HUT edge. We then add the utilizations of the identified HUT edges. The wirelength field is also computed by adding the wirelength fields of s_j with $c_{i,l}$. Next, in Algorithm 1, to apply Pareto minimization, we project each compound configuration in S^{tmp} to a two-dimensional point. Here, the first dimension is the utilization of the HUT edges, and the second dimension is wirelength. We ignore the resource utilization (which is wirelength excluding the infinite-capacity vias) to keep the number of dimensions small in order to reduce the runtime of the algorithm for the typically-large global routing instances. In the $reduce$ function, we also assume d_1 is the utilization of the HUT edges and d_2 is wirelength.

C. Routing Resource and Wirelength Minimization

In this variation of CGR, we consider minimizing wirelength as one objective and the routing resource usage as the second objective. The resource usage of each route is its wirelength excluding the vias on its path. This is because in this work, similar to others, we consider vias to have infinite capacity and hence they are not counted towards resource usage. So two routes can have the same wirelength but different resource usage, i.e., a difference in number of vias. When considering these two dimensions, a routing solution is better than another if it is not worse in wirelength and resource usage while it is better in at least one of these dimensions.

The reason we discuss this variation is because it shows the application of our (inherently multi-objective) framework for single objective wirelength minimization. This is because our second objective of minimizing the total routing resource usage is typically correlated with the total wirelength.

More specifically, in this variation, the configuration of each net has $2+R$ dimensions where the first two dimensions reflect the wirelength of the route and its resource usage, respectively. Resource usage is obtained by subtracting the number of vias from the wirelength. Similarly, the first two dimensions of the compound configurations reflect the wirelength and resource usage corresponding to the configurations (candidate routes) of the nets that were combined so far.

The $combine$ function in Algorithm 1 (line 8) simply adds the wirelength of compound configuration s_j with the wirelength of configuration $c_{i,l}$ of net i. The last R fields are also added element-wise to obtain s^{tmp}, and then s^{tmp} is added to the existing set $\mathcal{S}^{tmp}$. Next, to apply the min operation to $\mathcal{S}^{tmp}$ (line 10), first each compound configuration of $\mathcal{S}^{tmp}$ is projected to a two-dimensional point: the first

TABLE I
INFORMATION ABOUT THE ISPD 2008 BENCHMARK INSTANCES AND THE SOLUTIONS OF THE INPUT ROUTING PROCEDURES.

Benchmark	#Nets	Grid	# Layers	Vcap	Hcap	NTHU-R			BFGR			FastRoute		
						WL	OF	POW	WL	OF	POW	WL	OF	POW
adaptec1	176715	324x324	6	70	70	53.44	0	318043	54.31	0	332448	53.80	0	301465
adaptec2	207972	424x424	6	80	80	52.29	0	247900	52.32	0	283572	52.20	0	258564
adaptec3	368494	774x779	6	62	62	131.01	0	721996	131.41	0	788712	131.20	0	701841
adaptec4	401060	774x779	6	62	62	121.69	0	557546	121.63	0	660529	121.30	0	545372
adaptec5	548073	465x468	6	110	110	155.38	0	839255	156.72	0	934743	155.82	0	862231
newblue1	270713	399x399	6	62	62	46.53	0	198935	46.80	0	177140	46.30	0	211310
newblue2	373790	557x463	6	110	110	75.70	0	306686	75.70	0	347852	75.20	0	319470
newblue3	551667	973x1256	6	80	80	106.57	31454	545813	106.40	33900	589827	108.40	31276	553274
bigblue1	282974	227x227	6	70	70	55.95	0	358874	57.21	0	387691	56.60	0	370414
bigblue2	576816	468x471	6	52	52	90.59	0	414968	91.14	0	453963	91.20	0	407729
bigblue3	1122340	555x557	8	148	148	130.69	0	562362	131.79	0	648007	130.00	0	577168
bigblue4	2228903	403x405	8	204	204	230.90	160	998194	232.01	434	1007510	230.21	138	1024810
newblue4	636195	455x458	6	84	84	130.46	138	647181	130.80	218	740925	130.50	136	625358
newblue5	1257555	637x640	6	88	88	231.58	0	1122040	233.00	0	1266330	230.90	0	1171950
newblue6	1286452	463x464	6	132	132	176.89	0	924324	180.12	0	1008550	177.51	0	958422
newblue7	2635625	488x490	8	212	212	355.22	54	1643040	352.11	606	1827920	353.40	54	1631960

dimension is wirelength, and the second dimension is the aggregate resource usage (obtained by adding the last R fields of that compound configuration to get a resource usage value). Pareto minimization is then applied to find the Pareto points in this two-dimensional space. Finally, in the $reduce$ procedure, we assume d_1 is wirelength and d_2 is the resource usage.

V. SIMULATION RESULTS

We considered three input routing procedures from recent academic global routers: NTHU-Route 2.0 (denoted by NTHU-R) [4], BFGR [10] and FastRoute 4.0 (denoted by FastRoute) [30] in our first two experiments related to power and congestion. For the wirelength experiment we considered two additional routers, NCTU-GR [6] and MGR [29]. The solutions generated by NTHU-R and MGR were obtained from the corresponding authors (or the tool's website). The solutions of BFGR, FastRoute, and NCTU-GR were created by running their binaries and verifying the results using the ISPD 2008 evaluation script provided by the ISPD contest. Using these sets of solutions as candidate routes, we implemented our CGR procedure using C++ and conducted simulations using the ISPD 2008 benchmark instances [12]. The CGR procedure requires specifying one input solution as the reference case to adjust the edge capacities in its internal procedure. In our implementation we set the reference to be the input solution with the smallest total overflow.

Table I lists the information about each benchmark for the first three routers; columns 2 to 6 list the number of nets, grid granularity, number of layers, and the default vertical and horizontal edge capacities of each benchmark instance. Columns 7 to 15 list the corresponding wirelength (WL), overflow (OF), and power (POW) of each benchmark for each procedure. The wirelength is scaled to 10^5. We explain calculation of power in detail later. All simulations ran on a machine with a 2.8GHZ Intel CPU and 12GB of memory.

A. CGR for Minimizing Power and Wirelength

This section evaluates a variation of our CGR procedure for joint wirelength and power minimization as given in Section IV-A. In this experiment, we consider two cases. First, we assume a maximum wirelength degradation of 0%, compared to the Min-POW case which we explain shortly. We set $L = 4$ to explore the trade off between the wirelength and power as shown in Figure 5. In the second case, we allow up to 2% wirelength degradation to show that our framework can trade it off with improvement in power consumption. We use the input solution of the tool with minimum power for the net ordering procedure and to approximate the edge utilization in CGR's implementation.

To compute power consumption, first, for area capacitance, the relative wire sizes were scaled with respect to the wire size of layer M1 which we extracted from a 45nm technology library [19]. Then, to model coupling capacitance from the utilization of an edge in the grid-graph, we computed the remaining (not utilized) capacity on that edge by accounting for the wire sizes. We then computed the spacing between the wires on that edge, as given by Equation 6, and converted the spacing to coupling and fringe capacitance numbers by looking up the values for each metal layer from our considered 45nm technology library. The via edges were excluded in power computation because in the ISPD 2008 benchmarks vias are required to have infinite capacity which prohibited us to compute a wire spacing in Equation 6. After obtaining the capacitance values, to compute power, we assumed each net to have an activity factor which we randomly selected to be in the range of 0.05 to 0.2.

In our experiments, we compute the power of the routing solutions of the input routers. We note this is only to allow defining a base for comparison since these input solutions were not created with power as an objective. When creating the CGR routing solution, the procedure accounts for wire size and spacing as explained before. We note in both cases, i.e., when running the CGR procedure and when running the input routers (or using their solutions), the (normalized) capacity of each edge–which is the maximum number of routes that can pass from it without creating overflow–remain the same. Subsequently, the computation of overflow and utilization of an edge follows the same procedure for the input routers and CGR so their comparison is under the same conditions. The

TABLE II
RESULTS FOR MINIMIZING POWER AND WIRELENGTH

Benchmark	Min-POW			CGR											
				WL Degredation: 0%						WL Degredation: 2%					
					%imp. (Min-Pow)		POW %improvement				%imp. (Min-Pow)		POW %improvement		
	Tool	WL	POW	Time	WL	POW	NTHU-R	BFGR	FastRoute	Time	WL	POW	NTHU-R	BFGR	FastRoute
adaptec1	FastRoute	53.80	90560	2.82	0.67	8.69	8.74	15.44	8.69	2.13	0.19	9.96	10.01	16.62	9.96
adaptec2	NTHU-R	52.29	72681	2.04	0.21	7.41	7.41	19.02	11.25	2.64	-0.78	12.40	12.40	23.39	16.03
adaptec3	FastRoute	131.20	205830	5.75	0.16	4.87	7.49	15.32	4.87	5.83	-0.18	6.83	9.40	17.06	6.83
adaptec4	BFGR	121.60	163006	5.07	0.33	14.33	14.34	14.33	14.51	4.97	-0.85	17.98	17.99	17.98	18.16
adaptec5	NTHU-R	155.38	246487	6.00	0.44	8.94	8.94	18.22	11.38	5.69	-0.75	11.04	11.04	20.11	13.42
newblue1	NTHU-R	46.53	58450	2.42	0.39	5.79	5.79	17.50	11.31	2.39	-1.05	10.92	10.92	21.99	16.14
newblue2	NTHU-R	75.70	89601	3.41	0.33	14.86	14.86	24.94	18.29	2.42	-1.60	18.68	18.68	28.31	21.96
newblue3	NTHU-R	106.57	162609	5.30	0.42	12.67	12.67	18.01	14.45	5.35	-1.28	16.36	16.36	21.47	18.06
bigblue1	NTHU-R	55.95	105041	2.18	0.14	2.57	2.57	9.80	5.56	2.59	-0.46	4.32	4.32	11.43	7.26
bigblue2	NTHU-R	90.59	121562	3.56	0.00	14.19	14.19	21.55	16.74	3.60	-0.45	16.15	16.15	23.34	18.64
bigblue3	NTHU-R	130.69	165218	4.47	0.45	17.50	17.50	28.36	19.60	5.29	-1.32	20.65	20.65	31.10	22.67
bigblue4	FastRoute	230.20	291851	8.60	0.19	8.56	8.62	19.14	8.56	7.56	-1.46	14.31	14.37	24.23	14.31
newblue4	NTHU-R	130.46	189300	5.27	0.47	12.82	12.82	23.91	13.96	5.82	-0.85	16.53	16.53	27.14	17.62
newblue5	FastRoute	230.90	323494	5.81	0.03	4.24	5.84	16.55	4.24	9.13	-1.05	8.80	10.32	20.52	8.80
newblue6	NTHU-R	176.89	271056	7.37	0.27	5.73	5.73	13.56	9.06	6.64	-0.90	9.86	9.86	17.35	13.04
newblue7	NTHU-R	355.22	481342	6.39	0.06	12.41	12.41	21.28	13.92	8.46	-0.91	15.81	15.81	24.33	17.26
average				**4.06**	**0.29**	**9.72**	**10.00**	**18.56**	**11.65**	**5.56**	**-0.86**	**13.16**	**13.43**	**21.65**	**15.01**

only difference is that CGR also accounts for power within its internal procedure and takes in varying wire size and spacing numbers when calculating Equation 5.

Evaluation of power is the same in all the cases, i.e., CGR and the input solutions. Varying wire size and spacing values (which correspond to our technology library) are used to calculate Equation 5. Furthermore, the utilization of each edge which is used in calculation of the coupling capacitance is computed exactly according to the corresponding routing solution.

The results are reported in Table II. First, for each benchmark, the solution of three routing procedures which has the minimum power is designated as the Min-POW solution and reported in columns 3 to 4. The name of the routing procedure that provides the minimum power solution for each case is represented in column 2. The power for each benchmark for each procedure is reported in Table I. The results of CGR are reported in columns 5 to 16. Columns 6 and 7 report the percentage improvements in different metrics when we select the minimum power obtained from CGR, compared to the Min-POW case with the wirelength degradation threshold set to 0%. The power of CGR is improved on average by 9.72% compared to the Min-POW case and the wirelength is always less than or equal to that of the Min-POW case.

We also report the percentage improvement in power over each tool in columns 8 to 10. On average the savings in our power metric with respect to NTHU-R, BFGR, and FastRoute are 10.00%, 18.56%, and 11.65%, respectively. The overflow of CGR is at most equal to the overflow of the tool with the smallest total overflow which can be looked up from Table I.

The runtime of CGR is reported in column 5 for each benchmark. The runtime is on average 4.06 minutes.

For the case when up to 2% wirelength degradation is allowed, the results are reported from columns 11 to 16. In this case, the percentage of power improvement over the Min-POW case on average is 13.16%, while the wirelength on average is increased by 0.86%. For the benchmark `adaptec1` we improve both wirelength and power; the power improvement is larger than in the 0% degradation case. The improvements with respect to the base cases NTHU-R, BFGR, and FastRoute in this case are 13.43%, 21.65%, and 15.01%, respectively.

TABLE III
PARETO POINTS WHEN MINIMIZING POWER AND WIRELENGTH.

Benchmark	Pareto Points(% of Imp. Over Min-POW)							
	Point1		Point2		Point3		Point4	
	WL	POW	WL	POW	WL	POW	WL	Power
adaptec1	0.19	9.96	0.37	7.20	0.46	7.04	0.56	6.93
adaptec2	-0.78	12.40	-0.59	10.59	-0.21	9.80	0.36	9.09
adaptec3	-0.18	6.83	-0.09	6.14	-0.03	6.14	0.06	5.78
adaptec4	-0.85	17.98	-0.75	16.82	-0.65	16.40	0.00	16.05
adaptec5	-0.75	11.04	-0.69	10.68	-	-	-	-
newblue1	-1.05	10.92	0.06	10.32	1.14	8.79	0.62	7.99
newblue2	-1.60	18.68	-0.70	17.82	-0.53	17.75	0.00	0.00
newblue3	-1.28	16.36	-1.14	15.46	-	-	-	-
bigblue1	-0.46	4.32	-0.09	2.14	1.13	1.24	0.13	0.85
bigblue2	-0.45	16.15	0.76	14.92	1.53	14.30	0.21	13.68
bigblue3	-1.32	20.65	-0.41	19.94	-0.25	19.38	0.15	18.97
bigblue4	-1.46	14.31	-1.40	14.24	-	-	-	-
newblue4	-0.85	16.53	-0.66	15.48	-0.57	15.09	-	-
newblue5	-1.05	8.80	-1.04	8.16	-0.91	8.02	0.00	7.93
newblue6	-0.90	9.86	-0.89	9.07	-0.71	8.72	0.00	0.00
newblue7	-0.91	15.81	-0.91	15.48	-	-	-	-

In Table III, we also report the power and wirelength of all the Pareto points stored at the last stage of the CGR algorithm for the 2% WL degradation experiment. Each Pareto point represents a different routing solution obtained using our procedure. The entries are percentage improvement numbers relative to the wirelength and power of the Min-POW case. For example for benchmark `adaptec2` we have four Pareto points at the last stage of CGR, corresponding to four different global routing solutions. The first Pareto point results in 0.78% increase in wirelength with 12.40% decrease in power. The second Pareto point results in 0.59% decrease in wirelength with 10.59% decrease in power, and so on. Note that in benchmark `adaptec1`, all the Pareto points result in both wirelength and power improvements, compared to the Min-

TABLE IV
RESULTS FOR SIMULTANEOUS MINIMIZATION OF THE HUT EDGES AND WIRELENGTH

Benchmark	Min-HUT			CGR													
				L=4							L=7						
					%imp. (Min-HUT)			U_{HUT}	%improvement			%imp. (Min-HUT)			U_{HUT}	%improvement	
	WL	U_{HUT}	|HUT|	Time	U_{HUT}	|HUT|	WL	NTHU-R	BFGR	FastRoute	Time	U_{HUT}	|HUT|	WL	NTHU-R	BFGR	FastRoute
adaptec1	53.44	1273868	257960	1.65	3.67	1.71	0.17	3.67	12.75	6.78	2.65	3.67	1.71	0.17	3.67	12.75	6.78
adaptec2	52.29	1134378	201647	2.00	6.78	3.12	0.23	6.78	23.61	12.67	4.80	6.83	3.18	0.25	6.83	23.66	12.72
adaptec3	131.20	3233446	742605	4.95	1.99	7.96	0.26	5.74	14.92	1.99	8.58	4.03	7.98	0.27	7.70	16.69	4.03
adaptec4	121.30	2208114	637746	4.75	12.78	15.62	0.31	12.88	32.74	12.78	8.50	13.90	17.05	0.31	14.00	33.60	13.90
adaptec5	155.38	3496494	458074	5.80	5.60	2.04	0.21	5.60	18.05	9.92	12.41	8.20	1.89	0.21	8.20	20.30	12.40
newblue1	46.53	887098	198717	1.90	5.91	3.90	0.26	5.91	19.57	11.79	3.18	5.94	3.97	0.26	5.94	19.60	11.82
newblue2	75.70	1120624	218546	3.05	11.46	4.23	0.33	11.46	27.74	16.36	6.02	11.49	4.22	0.33	11.49	27.76	16.39
newblue3	106.57	2431870	465796	4.55	13.46	4.46	0.29	13.46	27.55	14.14	10.43	13.93	4.42	0.29	13.93	27.94	14.61
bigblue1	55.95	1611590	181535	2.05	1.75	0.10	0.11	1.75	8.99	6.79	4.86	9.33	0.07	0.11	9.33	16.01	13.98
bigblue2	90.59	1936104	495846	3.40	4.01	1.75	0.14	4.01	13.91	7.07	6.61	4.03	1.60	0.15	4.03	13.93	7.08
bigblue3	130.69	1923616	292576	5.05	14.08	6.69	0.41	19.13	33.95	19.13	14.52	18.96	9.93	0.41	23.72	37.70	23.72
bigblue4	230.90	3570538	378057	5.00	8.84	4.57	0.34	8.84	24.44	12.63	21.17	8.95	4.60	0.35	8.95	24.53	12.73
newblue4	130.50	2909278	520849	5.00	5.73	6.36	0.21	5.73	21.99	8.99	13.11	9.22	6.23	0.21	9.22	24.88	12.35
newblue5	231.58	4694444	821161	5.45	5.16	1.30	0.18	5.16	20.71	12.28	19.24	8.20	1.34	0.18	8.20	23.25	15.09
newblue6	176.89	3672558	395938	7.00	4.60	1.41	0.23	4.60	13.56	8.88	20.32	7.39	1.43	0.24	7.39	16.08	11.54
newblue7	355.22	6182088	596755	8.15	8.22	2.80	0.43	8.22	23.31	12.15	26.23	11.53	2.84	0.44	11.53	26.08	15.32
average				**4.36**	**7.13**	**4.25**	**0.26**	**7.68**	**21.11**	**10.90**	**11.41**	**9.10**	**4.53**	**0.26**	**9.63**	**22.80**	**12.78**

TABLE V
INFORMATION ABOUT THE HUT EDGES FOR THE SOLUTIONS OF THE INPUT ROUTING PROCEDURES.

Benchmark	NTHU-R		FastRoute		BFGR	
	U_{HUT}	|HUT|	U_{HUT}	|HUT|	U_{HUT}	|HUT|
adaptec1	1273868	257960	1316280	285338	1406310	295205
adaptec2	1134378	201647	1210836	219086	1384318	250004
adaptec3	3362124	714015	3233446	742605	3724838	830651
adaptec4	2210568	583105	2208114	637746	2863184	788055
adaptec5	3496494	458074	3664166	495728	4027622	527272
newblue1	887098	198717	946270	213842	1037758	230537
newblue2	1120624	218546	1186228	249327	1373000	273598
newblue3	2431870	465796	2431870	481585	2904822	618780
bigblue1	1611590	181535	1739818	196242	1698788	196805
bigblue2	1936104	495846	1999832	558911	2158884	536065
bigblue3	1923616	292576	2043724	312910	2502030	372292
bigblue4	3570538	378057	3570538	386102	4307742	445552
newblue4	3515696	497985	2909278	520849	2909278	587634
newblue5	4694444	821161	5075392	908105	5615164	964734
newblue6	3672558	395938	3844852	424234	4053064	442132
newblue7	6182088	596755	6182088	614273	7399204	696069
average	**2688979**	**422357**	**2722671**	**452930**	**3085375**	**503462**

POW case. The reason that the number of stored solutions is sometimes less than $L = 4$ is because after combining the compound configurations at the last stage of CGR, it is possible that infeasible configurations are eliminated and non-Pareto points are dominated. The removal of these configurations can result in the final number of stored solutions to be less than L. The availability of multiple solutions within one experiment provides designers another option to select appropriate solutions (besides the option to a-priori defining the WL threshold and overflow parameters).

B. CGR for Minimizing Congestion and Wirelength

In this section, we evaluate the variation of our CGR procedure for congestion spreading to be traded off with wirelength as given in Section IV-B. The objective of CGR is to simultaneously minimize wirelength and the total utilization of edges in the routing grid-graph which are utilized higher than 70% of their capacity. This threshold for highly-utilized edges is mentioned in [1]. We refer to these edges as the HUT edges (i.e., highly-utilized) and denote the summation of the utilization of the HUT edges by U_{HUT}. Reducing the utilization of the HUT edges allows more effective handling of issues such as crosstalk at the lower design stages [1]. Table V shows the information about the HUT edges for the different routing procedures. For each case, the first column represents the total utilization of the HUT edges and the second column shows the number of these edges. For each benchmark the underlined number shows the smallest U_{HUT} among the three different procedures. For example for `adaptec1`, NTHU-R has the smallest total utilization on the HUT edges.

We impose a bound of 0% for the maximum wirelength degradation so no wirelength degradation is allowed in this experiment. This is compared to the "Min-HUT" case which corresponds to one of the input solutions for each benchmark, as we shortly explain. We apply multi-objective CGR and among the stored solutions at the final step of the algorithm select the one with smallest U_{HUT} as the final solution. In CGR's net ordering procedure, the solution of the Min-HUT case is used.

In our framework the parameter L determines the maximum number of global routing solutions stored at each step of our procedure. Increasing L allows creating more solutions which provide a tradeoff set in the two considered objectives. However it also increases the runtime of our procedure. We experiment with two cases of L 4 and 7.

The results are reported in Table IV. For a routing solution, we report U_{HUT}, and the number of HUT edges which we denote by |HUT|. Consider columns 2 to 4 showing the Min-HUT case. In Min-HUT, for each benchmark, we identify the tool with minimum U_{HUT} among the three routing procedures based on the information in Table V. We report the values of WL, U_{HUT}, and |HUT|, for its routing solution.

We compare these metrics for the CGR solution in columns 5 to 11 for the case when $L = 4$. Columns 6 to 8 report

TABLE VI
RESULTS FOR WIRELENGTH MINIMIZATION FOR BENCHMARKS WITHOUT OVERFLOW

Benchmark	WL for Different Routers					CGR							
						M=4				M=5			
						L=5		L=10		L=5		L=10	
	NTHU-R	BFGR	FastRoute	NCTU	MGR	WL	Time	WL	Time	WL	Time	WL	Time
adaptec1	53.44	54.30	53.80	53.04	52.86	52.97	2.73	52.90	4.13	52.83	2.82	52.50	6.13
adaptec2	52.29	52.30	52.20	51.51	51.47	51.43	3.32	51.34	4.30	51.32	3.91	51.13	6.48
adaptec3	131.01	131.40	131.20	129.24	128.91	128.91	4.30	128.87	10.85	128.85	7.75	127.79	15.99
adaptec4	121.69	121.60	121.30	120.53	119.95	120.41	6.26	119.80	9.66	119.86	8.26	119.04	15.65
adaptec5	155.38	156.70	155.80	153.96	153.23	153.50	6.95	153.30	11.96	152.09	9.69	151.15	18.89
bigblue1	55.95	57.20	56.60	56.29	55.88	56.28	2.24	56.21	5.34	55.85	4.03	55.85	7.31
bigblue2	90.59	91.10	91.20	88.66	88.91	88.55	4.22	88.35	7.33	88.85	4.41	88.71	9.71
bigblue3	130.69	131.80	130.00	129.35	128.76	129.09	4.67	128.71	8.94	128.55	8.58	128.04	16.49
newblue1	46.53	46.80	46.30	45.76	45.63	45.45	3.01	45.27	4.96	45.57	3.25	45.37	6.34
newblue2	75.70	75.70	75.20	74.51	74.49	74.14	4.06	74.11	6.87	74.35	5.76	73.62	10.62
newblue5	231.58	233.00	230.90	228.68	228.01	228.15	7.44	228.03	14.40	227.83	11.23	225.90	26.83
newblue6	176.89	180.10	177.50	175.49	174.93	174.45	7.57	175.06	17.19	174.73	10.82	174.44	21.69
average	**110.15**	**111.00**	**110.17**	**108.92**	**108.59**	**108.61**	**4.82**	**108.50**	**8.58**	**108.39**	**6.63**	**107.80**	**13.25**

TABLE VII
WIRELENGTH MINIMIZATION FOR BENCHMARKS WITH OVERFLOW

Benchmark	WL and OF for Different Routers										CGR		
	NTHU-R		BFGR		FastRoute		NCTU		MGR		M=5 , L=10		
	WL	OF	WL	OF	WL	OF	WL	OF	WL	OF	WL	OF	Time
bigblue4	230.90	160	232.00	434	230.20	138	224.83	188	228.80	130	228.14	130	7.91
newblue3	106.57	31454	106.40	33900	108.40	31276	104.59	31688	105.79	31276	105.43	31276	5.72
newblue4	130.46	138	130.80	218	130.50	136	127.35	144	129.45	136	128.98	136	6.73
newblue7	355.22	54	352.10	606	353.40	54	342.37	54	349.94	58	341.48	54	15.78
average	**205.79**	**7952**	**205.33**	**8790**	**205.63**	**7901**	**199.79**	**8019**	**203.50**	**7900**	**201.01**	**7899**	**9.04**

the percentage improvement in these metrics in CGR over the Min-HUT case. On average, CGR reduces the utilization on the HUT edges by 7.13%, and the number of HUT edges by 4.25%. Interestingly, for these benchmarks, our framework provides solutions that are always better than the original solutions both in terms of wirelength and the congestion metrics. The average improvement in wirelength is 0.26%. Also note that the overflow of each edge does not exceed the overflow of the tool with minimum overflow. For the majority of the benchmarks (all except `newblue3`, `bigblue4`, `newblue4`, and `newblue7`), all the routing procedures have 0-overflow solutions (as can be seen in Table I) and no adjustment in capacity is necessary. So our solution has also 0 overflow. For example in `adaptec4` we have a 0-overflow solution while reducing |HUT| by almost 15.62% compared to the Min-HUT case and improving the wirelength by 0.31%.

Next, columns 9 to 11 report the percentage improvements in total utilizations of HUT edges in CGR compared to each tool. The U_{HUT} is improved by 7.68%, 21.11%, 10.90% in NTHU-R, BFGR, and FastRoute, respectively.

The runtime of CGR (in minutes) is reported in column 5. The average runtime of CGR in this experiment is 4.36 minutes. The runtimes remain consistently in the order of few minutes even for the unroutable benchmarks.

The results for the case when $L = 7$ are also reported in columns 12 to 18. Columns 13 to 15 show the percentage of improvement for U_{HUT}, the number of HUT edges, and wirelength over the Min-HUT case. On average the utilization of the HUT edges is reduced by 9.10% and the number of HUT edges by 4.53%. The wirelength is always less than the wirelength of the Min-HUT case. Finally, the percentage improvements in total utilizations over different tools are reported in columns 16 to 18. When $L = 7$, CGR improves the utilization of the HUT edges from NTHU-R, BFGR, and FastRoute, by 9.63%, 22.80%, and 12.78%, respectively. The runtimes for this case are larger than the case when $L = 4$, but they are still in the order of few minutes, consistently and even for the unroutable benchmarks. The average runtime in this case is 11.41 minutes. This increase in runtime is traded off with improvement in the solution quality; as can be seen U_{HUT}, |HUT| and WL are each improved in each benchmark compared to the corresponding case for $L = 4$.

In summary, in the congestion spreading experiment we showed that CGR can be used to significantly reduce the number and total utilization of the HUT edges with a runtime of few minutes for any of the ISPD 2008 benchmark and without any wirelength degradation.

C. CGR for Minimizing Wirelength

We implemented a variation of our CGR procedure for minimizing wirelength (together with routing resource usage) as described in Section IV-C. For the net ordering procedure of CGR, the input solution with the smallest total wirelength is used. The results are reported in Table VI for the routable benchmarks. Columns 2 to 6 report the wirelength of the five academic routers used to generate input routing solutions to CGR for this experiment.

TABLE VIII
LAYER-BASED CONTRIBUTION (IN PERCENTAGE) OF EACH TOOL IN adapect1

Tool	M1	M2	M3	M4	M5	M6
NTHU-R	54.66	50.64	60.32	60.95	58.43	57.77
FastRoute	39.90	36.27	22.87	21.33	9.19	8.46
BFGR	5.44	13.08	16.81	17.73	32.38	33.78

To verify the impact of the number of collaborating routers we experimented with $M = 4$ and $M = 5$. (For $M = 4$, MGR was excluded from the set.) To also show the impact of the number of routing solutions generated by CGR we experiment with $L = 5$ and $L = 10$. The combination creates four cases of CGR for this experiment. Table VI shows the wirelength (WL) and runtime for each case. The overflow is zero in CGR.

The multi-objective optimization in CGR is considering minimization of both wirelength and routing resource usage. In this case, a route with high wirelength may have low routing resource usage if many vias are used on its path. This is because vias are modeled with infinite capacity and hence not viewed as routing resource in our experiments. The consistent improvement in wirelength is observed even when both wirelength and routing resource usage are simultaneously optimized in this experiment.

As can be seen, the higher value of M and L each result in improvement in wirelength. Increase in each parameter increases the runtime of CGR. The maximum wirelength improvement is for $M = 5$ and $L = 10$ which results in an average runtime of 13.65 minutes over the benchmarks. In this case, the normalized wirelengths of MGR and CGR are 0.99X and 0.95X respectively.

Table VII shows the wirelength and overflow of all the tools. For CGR we only report the case when $M = 5$ and $L = 10$ which is the case resulting in the best solution quality. As can be seen, the normalized wirelength of MGR and CGR are 0.99X and 0.96X with the same overflow for each benchmark.

For a CGR solution, we can verify the amount and type of contribution by each collaborating tool. For example, we ran a simple experiment on adaptec1 for $M = 3$ using NTHU-R, BFGR, and FastRoute. For this benchmark, 60.12% of CGR wirelength is contributed by NTHU-R. FastRoute and BFGR have similar wirelength contribution of about 20%. However, we observed that the FastRoute contribution is spreading over the chip area, while the BFGR contribution is mostly concentrated in the middle-left and bottom-right of the chip. We verified that the BFGR contribution includes a higher number of shorter nets concentrated in these areas while the FastRoute contribution includes longer nets that span the chip area.

Table VIII shows the contribution of each routing tool per layer in this example. For each tool and metal layer a percentage is reported that corresponds to the degree of routing resource usage in the CGR solution in that metal layer. (So each column adds to 100%.) Here we observe that the contribution of BFGR increases when moving to the higher metal layers (from 5.44% in M1 to 33.78% in M6), while the contribution of FastRoute decreases (from 39.90% in M1 to 8.46% in M6). The contribution of NTHU-R is maximum in M3 and M4.

VI. DISCUSSION AND CONCLUDING REMARKS

We introduced CGR as a procedure for collaborative multi-objective global routing which receives as input different routing solutions. In general, for tight resource constraints, the CGR procedure may fail to find a feasible solution. However, we showed in practice a feasible solution is always attainable in our experiments. Furthermore, significant improvement is possible to optimize for other objectives in addition to the primary objectives of wirelength and overflow used in global routing. Obtaining feasible and high-quality solutions in our experiments is due to effective realization of the Pareto-algebraic procedure of CGR for the three presented global routing variations. Key factors contributing to effective realization are summarized below.

First, in the CGR formulation, the capacity of each edge is always adjusted according to the overflow of the input solution with the *smallest* total overflow. This helps controlling overflow even though it is not explicitly considered in the CGR procedure.

Second, due to the values of the edge capacities in the global routing benchmarks (which are often much higher than 1), infeasibility won't be an issue in the initial iterations of the CGR procedure because the edges are typically not utilized to their capacities.

Third, we apply a net ordering procedure which starts processing the nets passing from the congested areas first. These areas contain more overflow edges and by processing them prior to the un-congested regions, we allow for the maximum flexibility to avoid infeasibility in the later iterations. Moreover, we observed that the routes generated by the input routers were more diverse for the nets passing from the congested areas and not much different in the un-congested areas. This observation also helped with achieving feasibility combined with our net ordering procedure.

Fourth, a cost function is reasonable if it behaves monotonically with respect to the dimensions of a configuration. Since all the considered cost functions in this work are monotonic, in the exact version of CGR (when the $reduce$ procedure is excluded), a dominated configuration is guaranteed not to yield any better solution in future iterations and can be removed with certainty. Consequently, the Pareto points allow capturing the potentially interesting configurations.

Fifth, in the *reduce* phase, we select L configurations which provide a good approximation of the current configuration set. We choose configurations that cover the entire Pareto curve range uniformly. This method ensures that we always store the solution with *the lowest resource usage* as well as the one with the lowest cost and in addition some intermediate ones. This sampling method provides the highest possible flexibility in selecting fitting configurations in later steps.

When considering the selection of parameters M and L, available memory is a factor given the large size of the ISPD 2008 global routing benchmark which reflect industry-sized instances. Specifically, for each net M configurations

are stored. At most L distinct routing solutions are stored at the end of each iteration of CGR. Furthermore, the number of intermediate stored solutions may exceed L when combining the configurations of a new net with the current configuration set. Moreover, the increase in parameters M and L increases the runtime of the algorithm; however it also allows more improvement in the solution quality as illustrated in the simulation results. In practice we expect M, the number of collaborating input solutions, to be a small value similar to those specified in our experiments. The parameter L can be set based on the memory size for the considered benchmark as well as the user's choice to tradeoff the solution quality with runtime of the procedure.

REFERENCES

[1] C. Alpert and G. Tellez. The importance of routing congestion analysis. In *Design Autom. Conf. (Knowledge Center)*, 2010.

[2] L. Behjat, A. Vannelli, and W. Rosehart. Integer linear programming models for global routing. *INFORMS Journal on Computing*, 18(2):137–150, 2006.

[3] Y.-J. Chang, T.-H. Lee, and T.-C. Wang. GLADE: A modern global router considering layer directives. In *Int'l Conf. on Computer-Aided Design*, pages 319–323, 2010.

[4] Y.-J. Chang, Y.-T. Lee, and T.-C. Wang. NTHU-Route 2.0: a fast and stable global router. In *Int'l Conf. on Computer-Aided Design*, pages 338–343, 2008.

[5] M. Cho, K. Lu, K. Yuan, and D. Z. Pan. Boxrouter 2.0: A hybrid and robust global router with layer assignment for routability. *ACM Trans. on Design Autom. Electr. Syst.*, 14(2), 2009.

[6] K.-R. Dai, W.-H. Liu, and Y.-L. Li. NCTU-GR: Efficient Simulated Evolution-Based Rerouting and Congestion-Relaxed Layer Assignment on 3-D Global Routing. *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 20(3):459 –472, 2012.

[7] M. Geilen and T. Basten. A calculator for Pareto points. In *Design, Autom., and Test in Europe*, pages 16–20, 2007.

[8] M. Geilen, T. Basten, B. D. Theelen, and R. Otten. An algebra of Pareto points. *Fundam. Inform.*, 78(1):35–74, 2007.

[9] C.-H. Hsu, H.-Y. Chen, and Y.-W. Chang. Multilayer global routing with via and wire capacity considerations. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 29(5):685–696, 2010.

[10] J. Hu, J. A. Roy, and I. L. Markov. Completing high-quality global routes. In *Int'l Symp. on Physical Design*, pages 35–41, 2010.

[11] ISPD 2007 Contest [Online]. In *http://archive.sigda.org/ispd2007/*.

[12] ISPD 2008 Contest [Online]. In *http://archive.sigda.org/ispd2008/*.

[13] T.-H. Lee, Y.-J. Chang, and T.-C. Wang. An enhanced global router with consideration of general layer directives. In *Int'l Symp. on Physical Design*, pages 53–60, 2011.

[14] T.-H. Lee and T.-C. Wang. Congestion-constrained layer assignment for via minimization in global routing. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 27(9):1643–1656, 2008.

[15] M. D. Moffitt. Maizerouter: Engineering an effective global router. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 27(11):2017–2026, 2008.

[16] M. D. Moffitt, J. A. Roy, and I. L. Markov. The coming of age of (academic) global routing. In *Int'l Symp. on Physical Design*, pages 148–155, 2008.

[17] M. Moser, D. Jokanvi, and N. Shiratori. An algorithm for the multidimensional multiple-choice knapsack problem. *IEICE Trans. on Fundamentals of Electronics*, E80-A(3):582–589, 1997.

[18] D. Müller. Optimizing yield in global routing. In *Int'l Conf. on Computer-Aided Design*, pages 480–486, 2006.

[19] Nangate 45nm Open Cell Library [Online]. In *http://www.nangate.com*, 2008.

[20] M. M. Ozdal and M. D. F. Wong. Archer: A history-based global routing algorithm. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 28(4):528–540, 2009.

[21] J. A. Roy and I. L. Markov. High-performance routing at the nanometer scale. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 27(6):1066–1077, 2008.

[22] R. Shelar and M. Patyra. Impact of local interconnects on timing and power in a high performance microprocessor. In *Int'l Symp. on Physical Design*, page invited talk http://ispd.cc/slides10/8_01.pdf, 2010.

[23] H. Shojaei, A. Davoodi, and J. T. Linderoth. Congestion analysis for global routing via integer programming. In *Int'l Conf. on Computer-Aided Design*, pages 256–262, 2011.

[24] H. Shojaei, A. Davoodi, and P. Ramanathan. Confidentiality preserving integer programming for global routing. In *Design Autom. Conf.*, pages 709–716, 2012.

[25] H. Shojaei, A. H. Ghamarian, T. Basten, M. Geilen, S. Stuijk, and R. Hoes. A parameterized compositional multi-dimensional multiple-choice knapsack heuristic for CMP run-time management. In *Design Autom. Conf.*, pages 917–922, 2009.

[26] H. Shojaei, T.-H. Wu, A. Davoodi, and T. Basten. A Pareto-algebraic framework for signal power optimization during global routing. In *Int'l Symp. Low Power Electronics and Design*, pages 407–412, 2010.

[27] T.-H. Wu, A. Davoodi, and J. T. Linderoth. A parallel integer programming approach to global routing. In *Design Autom. Conf.*, pages 194–199, 2010.

[28] T.-H. Wu, A. Davoodi, and J. T. Linderoth. GRIP: Global routing via integer programming. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 30(1):72–84, 2011.

[29] Y. Xu and C. Chu. MGR: Multi-level global router. In *Int'l Conf. on Computer-Aided Design*, pages 250 –255, 2011.

[30] Y. Xu, Y. Zhang, and C. Chu. Fastroute 4.0: global router with efficient via minimization. In *Asia South Pacific Design Autom. Conf.*, pages 576–581, 2009.

[31] M. Yukish. Algorithms to identify Pareto points in multi-dimensional data sets. *PhD Thesis, Pennsylvania State University*, August 2004.

Hamid Shojaei received his M.S. degree in Computer Engineering from the University of Tehran in Iran and Ph.D. degree in Electrical and Computer Engineering from the University of Wisconsin - Madison in 2004 and 2012, respectively. His research interests include electronic design automation for modern very large scale integrated (VLSI) circuits, design verification, and embedded systems. Hamid has joined Qualcomm Inc. since January 2012.

Azadeh Davoodi received the Ph.D. degree in Electrical and Computer Engineering from the University of Maryland, College Park in 2006. Currently, she is an Associate Professor with the Department of Electrical and Computer Engineering, University of Wisconsin - Madison. Her research interests are in various areas of Electronic Design Automation (EDA) including security in cloud-based EDA, and optimization to improve scalability, reliability, power and performance in current and future IC design paradigms. Azadeh is recipient of a 2011 National Science Foundation CAREER award.

Twan Basten is a professor in the Electrical Engineering department at Eindhoven University of Technology and a research fellow of the Embedded Systems Institute, both in the Netherlands. He holds an MSc and a PhD in Computing Science from Eindhoven University of Technology. His research interests include embedded and cyber-physical systems, dependable computing and computational models.