Orthogonal Greedy Coupling - A New Optimization Approach to 2-D FPGA Routing

[*]Yu-Liang Wu [+]Malgorzata Marek-Sadowska

[*]Cadence Design Systems, Inc. San Jose, CA 95134
[+]Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106.

Abstract

We propose a novel optimization scheme that can improve the routing by reducing a newly observed router decaying effect. A pair of greedy-grow algorithms, each emphasizing a different optimization target are designed. By applying one algorithm first and then switching to the other when the first one approaches its decaying stage, the undesired effect can be significantly reduced and thus better results are produced. On the tested MCNC and industry benchmarks, in addition to our very low segment consumption the total number of tracks used by our scheme is 37% less than a published conventional maze router and 22% less than the best known 2-step global/detailed router [4,5].

Our results show that complicated multi-objective problems could be effectively attacked by coupling low complexity algorithms that traverse the solution space in orthogonal directions. This idea is applicable on both algorithmic and architectural optimization approaches [7].

1. Introduction

Due to the fast growing rate of today's Integrated Circuit size, low complexity algorithms are desirable for many design automation applications. However, as most of today's CAD problems are complicated intractable problems, applying simple heuristics yields suboptimal results. In this paper we propose an effective routing optimization scheme using only low complexity greedy algorithms to achieve near optimum routing results for a 2-D Xilinx 4000-like FPGA architecture [1,3,4,5,7-10] (see Fig. 1, 2).

Routing is crucial in FPGA design automation since feasibility of a design is more constrained by the programmable routing resources than by the logic gates [2]. Minimizing the number of routing tracks and routing segments are the two objectives of today's FPGA routers. However, we show that even achieving just one of these goals is an NP_hard problem. These two objectives are related but are not identical. After the router has made an effort to minimize the number of used tracks, pushing this objective further may cause a sharp increase of the utilized routing segments, and vice versa. Methodologies attacking both targets at the same time have been rarely proposed in the past.

The routing architecture considered [7,8,9,10] is composed of disjoint domains. Therefore one view of the routing problem is a 2-dimensional (2-D) generalization of the classical channel routing problem with no vertical constraints. The other analogy is to the bin packing problem. Besides the maze router, two types of deterministic algorithms have been proposed: the global/detailed 2-step router [4,5] and the one-step router [9] which makes explicit use of the disjoint domain property. However, analyzing the results of both heuristics one can observe an undesirable phenomenon that the few last routed track domains are very sparsely populated. Additionally, both heuristics produce results which are difficult to improve even when a costly rip-up and reroute techniques are applied or extensive tuning of program parameters is performed. We conjecture that this phenomenon is hard to avoid if greedy algorithms are applied in a conventional way.

We propose a new philosophy in applying greedy strategies to approach these difficult to solve NP-hard routing problems. Our target will be to eliminate the undesired effect of poor utilization of the last routed track domains. We conjecture that since our problem does not have the optimal substructure property (e.g. matroids), a greedy algorithm which applies one cost function may traverse the solution space only along a particular direction. After applying such a single cost function for a while, the quality of the produced results starts to deteriorate rapidly, which leads to a premature local minimum. This behaves as if the resources available for the effective operation of that specific heuristic were depleted therefore triggered a sudden quality drop in the final stages. We also conjecture that most of the carefully designed heuristics perform nearly equally well at the early stages of their execution process since they all capture the major optimization factors of the problem. However, while reaching the depletion stages, switching to another heuristic with most orthogonal traversal path in solution space would better prevent a premature conversion. We will apply these conjectures in designing a router for our FPGA architecture.

Based on our problem formulation, we derive a generalized greedy-grow principle encapsulating both bin-packing and 2-D channel routing models. The two greedy-grow routing algorithms are designed in a naturally complementary way: one grows continuously from a well-picked single seed and the other grows discretely from multiple seeds. Each of them chooses a sound but different priority in routing path selections. Each of the algorithms achieves good results comparable to previously published deterministic router results for this architecture [4,5,9]. However, similar to all the other greedy heuristics we

32nd ACM/IEEE Design Automation Conference ®

experimented on this architecture, these two algorithms are also not exempt from the track utilization decaying effect in their later stages, even after substantial tuning of program parameters. We have also tried to combine the major objectives of both algorithms into a single cost function as well as to alternate these two algorithms on odd and even track domains. The results were quite similar: in each case we have observed the sharp quality degeneration phenomenon in near completion stages.

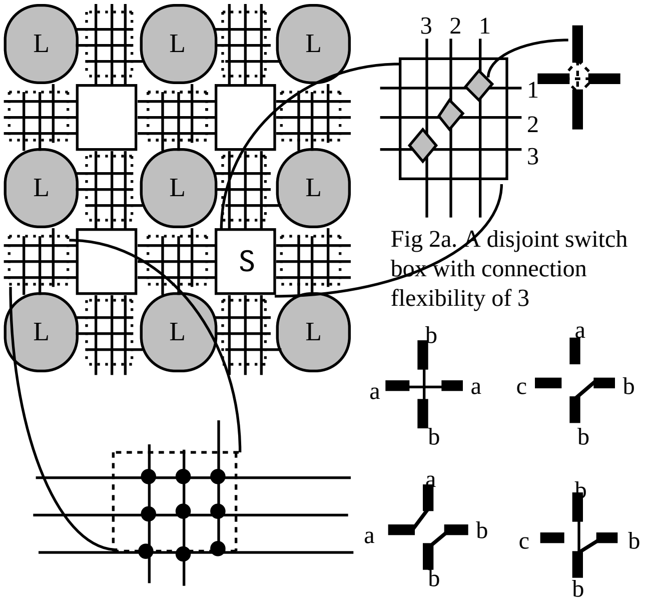

Fig 2a. A disjoint switch box with connection flexibility of 3

A connection (C) box with full connection flexibility

Fig. 1. 2-D FPGA routing architecture

Fig. 2b. The switch (S) box connection examples of each intersection "diamond" shown above

But, when the algorithms are coupled - one routes approximately the first half of the track domains (typically 7 in our experiment), and the other routes the remaining nets on the other track domains - the track utilization results are suddenly improved.

Our experimental results seem to support very well the usefulness of this new optimization technique. On the tested MCNC and industry benchmarks [4,5], the total number of tracks used by our scheme is 37% less than a published conventional maze router and 22% less than the best known 2-step global/detailed router based on the same pin assignments. Besides, in our results, only two circuits are routed with their average 2-pin connection segment consumption 2% over their average minimum rectilinear path. Our method can also be adapted to handle performance driven routing in a simple and efficient way. Our router can be linear on the number of nets in both CPU time and run-time memory which also suggests its suitability for very large circuits.

We believe that this novel methodology, although quite simple, could suggest a new approach in solving many hard combinatorial and CAD problems where optimal polynomial solutions are unfeasible and a quick effective method is desired.

The paper is organized as follows. In section 2 we introduce the routing architecture and in section 3 we discuss the previous work. In section 4, we formulate the problem and show its computational complexity. In section 5, we discuss in more detail the FPGA track utilization decaying effect. In section 6, we introduce the notion of orthogonal greedy coupling and present details of our approach. In section 7 we show the experimental results In section 8 we give conclusions.

2. Routing Model and Terminology

The investigated architecture is depicted in Fig. 1. It is a 2-D array of look-up-tables (L), *connection* (C) *boxes* and *switch* (S) *boxes* [1,5,7], where all routing boxes of the same type have the same connection topology. In this *unit-segmented* architecture, all wire segments are of unit-length that span between two adjacent S boxes and run in both vertical and horizontal channels with the same number of tracks W. In each channel, tracks are assigned tracks *ids* numbering from right to left and from top to bottom. The C boxes contain routing switches that can be programmed to connect logic pins to wire segments. Here, we assume a logic pin in any C box can connect to any segment running through it. The S boxes contain switches that allow one wire segment to be connected to any adjacent wire with the same track id. Therefore, this routing architecture is *disjoint,* since when all S box switches are closed, then all wire segments are partitioned into D>1 disjoint connected wire sets, called *domains*, and a given domain will cover exactly one track in every C box. A 2-pin net connection can only occupy segments of the same track id across the whole chip.

3. Previous Work

In the past, for the 4000 Xilinx architecture, besides the conventional maze routing approach, the majority of the published FPGA routers follow a 2-step global/detailed routing scheme. The most popular among them is probably the graph search based CGE router [4,5], which is also well designed to experiment with various switching flexibilities for routability justification. The other kind is a one-step constrained bin-packing router (GBP) [9], which makes an explicit use of the disjoint routing domain property.

In the earlier work [7,8], it has been shown that in this specific architecture, the global to detailed routing mapping problem is NP-complete, and a router designed based on 2-step approach may exhibit a pathological phenomenon that for some examples global routed with less channel density may require more tracks than when globally routed with larger channel density. This phenomenon has been referred to as *Mapping Anomaly* in [9] which can not be detected by a just localized routability checking. On the other hand, the GBP router adopts a one-step scheme that only makes routing on one domain at a time. In this scheme, all disjoint routing domains are treated as independent packing bins of the same capacity, and each net is considered as a variable size object to be packed. By using a variant of the Best Fit Decreasing (BFD) [6] heuristic, satisfactory results have been consistently achieved, probably because this approach is less affected by the counterintuitive mapping anomaly.

However, although both routers can achieve substantial improvements over the conventional maze router, an adverse phenomenon, which we call the greedy decaying effect, is commonly observed on results produced by both routers.

4. Problem Complexity Analysis

In a practical routing task, completing the routing in the exactly minimum number of tracks is not necessarily the most important objective. The routing tracks are pre-fabricated and as long as the routing can be accommodated, the next goal is to improve the performance of routing. The number of segments (and therefore the number of switches) consumed by routing directly affects the active interconnection capacitance, this in turn affects the performance of the whole chip. Here, we consider the objective of completing the routing, given pin positions, in the minimum number of segments within the specified track size limit. Below we formulate the problems and analyze their computational complexity.

Problem: [Routing Problem (RP)]

Input: A 2-D FPGA chip with W tracks, pin positions of a set of multi-pin nets produced by the placer.

Decision problem (DRP): Can this routing be completed in W tracks?

Optimization problem (ORP): Finding the minimum number of tracks to complete the routing.

Problem: [Routing with Segment Usage Bound (RSB)]

Input: A 2-D FPGA chip with W routing tracks, pin positions of a set of multi-pin nets produced by the placer.

Decision problem (DRSB): Can the routing be completed in W tracks with W*S segments? S is the number of segments in a routing track.

Optimization problem (ORSB): Finding the minimum number of routing segments required to complete the routing in specified number of tracks.

Computational complexity of the above problems is given by the following theorems [7]. Here we skip the proofs.

Theorem 1. Both [DRP] and [ORP] are NP-hard for unit-segmented architectures.

Theorem 2. Both [DRSB] and [ORSB] are NP-hard for unit-segmented architectures.

5. The Greedy Decaying Effect

We define the *greedy decaying effect* as the dramatic performance drop phenomenon in the near completion stage when a greedy algorithm is applied to a complicated problem possessing no optimal polynomial solutions (e.g. an NP-hard problem). Here we mention two experiments to illustrate this notion.

We have run the CGE [4] router on a number of examples in the following scenario. For the i-th example, let W_i be the number of required tracks to complete the routing. We have run each of the examples with the available number of tracks being W_i - k, for k=1,2,..6. Fig. 3a. shows how many unrouted nets were created by this process. The gradual decrease of W_i causes, as expected, an increase of unrouted nets. However, it is quite interesting and surprising to observe that reduction of the W_i by up to a few tracks causes only very few unrouted nets, regardless of the circuit size or net population. For example, although circuit alu4 contains 851 2-pin nets, which can be completely routed in 15 tracks by CGE; however, by 14 tracks only 1 net is left unrouted (We might have expected this to be some number near 851/15.). The router can be viewed to always perform very poorly in near completion stages.

W | nets | unrouted	W_i	W_{i-1}	W_{i-2}	W_{i-3}	W_{i-4}	W_{i-5}	W_{i-6}
cht	0	1	2	3	5	6	6
alu4	0	1	(10?)				
example2	0	1	3	5	(7?)		
too_large	0	1	49				
apex7	0	1	2	4			
vda	0	(12?)	(23?)				

(?) : the number of unrouted nets seen in a non-stopped ripup&re-routing
Wi: the number of tracks required to have all nets routed by CGE
(Inside shown are the number of un-routable nets when the corresponding tarck sizes W are reduced)

Fig. 3a. Greedy decaying effect in CGE 2-step router

A similar behavior is also observed on the results produced by the GBP router [9]. In Fig. 3b we show the percentage of segments used (*segment usage*) in each domain for a completed routing of circuit alu4. It is evident that the effectiveness of this router decays rapidly for the last few domains, although the early domains are packed very efficiently. This behavior is quite common in both GBP and CGE results.

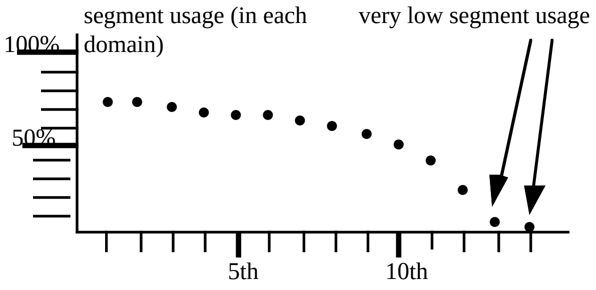

Fig. 3b. A typical greedy decaying effect in Bin-Packing Router (circuit: alu4)

A similar phenomenon can also be observed in applying a greedy heuristic for the NP-complete graph coloring problem. Regardless of the graph sizes and greedy objectives, only very few nodes are colorable by the last few applied colors, compared to those earlier applied colors. Intuitively, we suspect that as a greedy process can not perfectly fit in a problem possessing no optimal substructure property, after it is applied for a while, the left solution resources will eventually be forced to develop a somewhat "deformed configuration"[1] which causes the algorithm to be trapped in a premature local minimum. As a different greedy objective would develop a different deformed configuration, it should be less affected by one developed by another heuristic, especially those with an orthogonal coupled objective. This is the philosophy we based upon in developing our orthogonal coupling router.

6. Orthogonal Greedy Coupled Routing

[1] In our routing cases, it could be an extremely irregular left unrouted pin topology.

To make a further improvement possible, we propose a novel approach to eliminate this undesired behavior of 2-D FPGA routers.

Here, we informally define the *Orthogonal Coupled Algorithms* as two algorithms, such that:

Each of them is sound, captures major properties of the optimized problem and applies a different optimization strategy from that of the coupled algorithm and

The coupled algorithm is able to route the remaining nets effectively when the leading algorithm is approaching its depletion stages.

We will design the coupled algorithms using the principle of the greedy-grow explained in Fig 4 below. It can be seen that the classic left-edge routing algorithm [14], which always picks the left most seed net in each domain, is a 1-D version of this greedy-grow principle.

```
Greedy-Grow Principle

Loop until all nets are routed (or all domains are full)
    for each new routing domain
        • pick a seed net ;
        • always pick the next net that will
          cause the least resource waste,
          e.g. the net most adjacent to the routed segments
```

Fig. 4. Greedy-Grow Principle of disjoint domain routing

In Fig. 5, we just highlight the major conceptual difference of our orthogonal greedy coupled algorithms, which are jointly designed to attack the greedy decaying effect and the multiple routing objectives of minimizing routing tracks and segments (more details are shown at [7]). Note, that the multi pin nets are decomposed into 2-pin nets.

GG_S	GG_M
single-seed grow	multiple-seed grow
continuous grow	allows discrete grow
does not take into account pin congestion	takes into account pin congestion

Fig. 5. Comparison of our two orthogonal coupling greedy algorithms

Both algorithms honor the basic greedy-grow principle on disjoint domains, and each one routes one domain at a time to avoid the Mapping Anomaly. They differ in the strategy of deciding net ordering and routing paths, i.e. the way they greedily grow from current routed segments in a current domain. Each algorithm has a different explicit and implicit goals in routing. The explicit goal of GG_S is to reduce the number of routing segments with an implicit effect of reducing the number of tracks. GG_M has as its major objective the minimization of the used track domains with a secondary goal of minimizing the number of wire segments.

Since the nets are decomposed, they may share pins if they are sub nets of some originally multi pin net. In GG_S, a net whose pin inside a C box is already connected to a wire on the currently processed domain, will be given a higher score. Otherwise, if the net's pin is located in a C box adjacent to the wire segments used in the processed domain, an extra score of 1 is given. In case a net is already trapped, i.e. it has only one path to route, a high score is added. Each time after a net is routed, the information regarding the unrouted net count of all C boxes is updated, and the net with pins inside the most congested C boxes will be given an additional score of 1. Finally, the net is pseudo-routed, and if any segments can be saved, due to free segment sharing with earlier partial routes of the same net, this information also enters the final score. In our current implementation, when selecting a net to be actually routed, all the remaining nets are pseudo-routed, and the one with the highest score is selected. Therefore, this is an O(N^2) process, where N is the total number of nets. (But since the range of possible scoring is a bounded constant, this process can be implemented in O(N) without difference in results). The basic objective of the GG_S is to greedily absorb nets which have one pin attached to the existing routing (or which are the closest) and with more segment sharing available. Therefore, it follows a Kruskal like Rectilinear Steiner Tree algorithm for multi-pin nets while still maintaining low channel density. In this greedy strategy the channel density is not explicitly enforced, however, since this algorithm has strong capability of routing all nets in the near minimum number of segments, the final number of tracks is also implicitly reduced. Table 2 shows the results.

The second heuristic, GG_M, applies very strict rules in suppressing the increase of used domains while still controlling net lengths. The path selection strategy here is quite similar to that of the GBP router. Since it doesn't give preference to neighboring nets, instead, favors the nets with a minimum path length which do not pass through the congested area, the router behaves as growing connections from multiple seeds.

6.1. The Coupling Effects

In Table 1, the segment utilization of each domain is shown for 3 circuits when routed by the GG_S, GG_M, and the coupled heuristic GG_S@GG_M. In the coupled strategy, GG_S is applied on the first 7 domains, and GG_M works on the remaining domains.

When any of the greedy routers, GG_S or GG_M, is applied alone, they can achieve good results in comparison to all previously published deterministic routers for the considered architecture. However, each of them experiences the sharp decaying effect, which is shown in their "cosine wave" like deteriorating ending curves of segment utilization (Fig. 6b) and extremely low segment usage in the last few track domains. But, when they are coupled, the second algorithm shows its better immunity to the greedy decaying effect generated by the first algorithm in raising the ending curves. The mitigation of this decaying effect is also indicated by the higher segment usage in the last domain (see Fig. 7).

The coupled scheme can also be viewed as a simple while effective rip-up and reroute method. If the routing goal has not been met, then the domain where the GG_M is for the first time applied can be gradually moved until better results are reached.

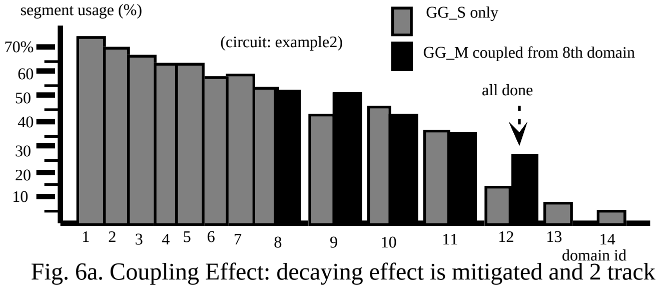

Fig. 6a. Coupling Effect: decaying effect is mitigated and 2 tracks are reduced

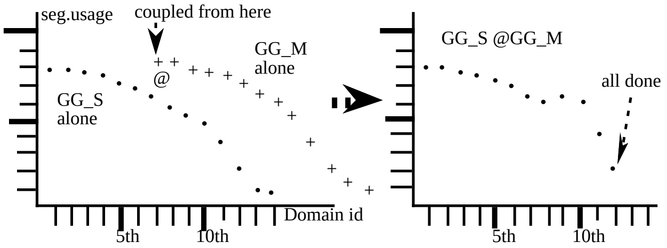

Fig. 6b. A conceptual view of the Coupling Effect (the dangling decaying curve, occuring when each algorithm is applied alone is eliminated when two are coupled.)

7. Experimental Results and Analysis

In Table 2 we show routing results of 5 typical schemes. In addition to the number of used tracks, we also show the average segments consumption per 2-pin net in the "av.seg" column, which is the value of (all segments used - summation of minimum lengths of all 2-pin nets) / (summation of minimum lengths of all 2-pin nets). This value can be negative due to segment sharing between same signal 2-pin nets. Our experiment show that not any algorithm pair can always produce better results, maybe because their cost functions are not "orthogonal" enough. And it is not true either that a more greedy heuristic will always bring results better than a less greedy one. Here we just briefly mention some other strategies and their major results.

• GG_S/GG_M (alternating after every routing domain, GG_S on odd domains and GG_M on even domains): Since switching 2 algorithms too often would behave like applying a single algorithm, some circuits still show the decaying effect.

• GG_SM: The major objectives of both greedy strategies are combined into one greedy, i.e. this greedy adopts the scoring scheme of GG_S and path choice of GG_M. The results are a little surprising. As shown in the Table 2, they are less stable compared to the coupled results. Some are showing an even worse decaying effect, e.g. in circuit k2, the summation of the segment usage of the last 4 tracks is just 6.33%! (3.10%, 2.15%, 0.60%, 0.48%). Overall, its performance is not better than any of the less greedy algorithms. Maybe, just because it is too greedy it suffers from its own decaying effect.

• Other variant coupling scheme: Instead of picking the best fit in each round of GG_M, we just pick the first fit. The results are nearly the same as with the original strategy.

The results clearly demonstrate the effectiveness of using well coupled simple heuristics in solving complicated problems.

8. Conclusions

In this paper, we addressed the phenomenon of a greedy decaying effect which 2-D FPGA routers commonly exhibit. We show in our experiments that this problem often causes many reasonable or overly greedy algorithms (e.g. GG_SM) to become trapped prematurely in a local minimum on their searching path, and this problem is hard to alleviate using conventional schemes.

In order to attack both routing objectives of minimizing routing tracks and segments and at the same time to avoid the undesired greedy decaying effects, which appear in all known to us routers for the considered FPGA architecture, we propose a non-conventional optimization scheme: the orthogonal greedy coupling. In this novel scheme, two equally effective algorithms are coupled in a way that each has a different optimization objective, the second one takes place after the first one has depleted enough its effectiveness. This approach is shown to be surprisingly effective to our routing problem. For a problem containing more than 2 optimization objectives, the methodology could be extended to include more than 2 coupled heuristics. We hope this philosophy could also be useful in attacking other hard combinatorial and CAD problems.

9. Acknowledgments

The authors would like to thank Professor J. Rose and Mr. Zewa Lao from the University of Toronto for supplying us with the placed circuits used in our routing experiments. This work was partially supported by the NSF grant MIP91-17328 and partially by AT&T and Xilinx through the California MICRO program.

References:

[1]. H. Hsieh, et. al. "Third-Generation Architecture Boosts Speed and Density of Field Programmable Gate Arrays", *Proc. CICC,* pp. 31.2.1-31.2.7. 1990.

[2]. S. Trimberger and M. Chene, "Placement-Based Partitioning for Lookup-Table-Based FPGA's", *Proc. ICCAD,* pp. 91-94, 1992.

[3]. M. Palczewski, "Plane Parallel A* Maze Router and Its Application to FPGAs", *Proc. of DAC*, pp.691-697, 1992

[4]. S. Brown, J. Rose, and Z. G. Vranesic, "A Detailed Router for Field-Programmable Gate Arrays", *IEEE Trans. on Computer-Aided Design,* Vol. 11, No. 5. pp. 620-628, May 1992.

[5]. Guy G. Lemieux and Stephen D. Brown, "A Detailed Routing Algorithm for Allocating Wire Segments in FPGAs", *4th ACM/SIGDA Physical Design Workshop*, 1993.

[6]. Johnson, Demers, Ullman, Garey and Graham, "Worst-Case Performance Bounds for Simple One-Dimensional Packing Algorithms", *SIAM Jr. On Computing,* 3(4), pp.299-325 (1974).

[7]. Y.L. Wu, "On 2-D FPGA Routing: Theoretical Analysis and Novel Effective Solutions", *Ph.D. Thesis, Dept. of Electrical and Computer Engineering, UCSB*, 1994.

[8]. Y.L. Wu, S. Tsukiyama, and M. Marek-Sadowska, "Graph Based Analysis of 2-D FPGA Routing", submitted for publication.

[9]. Y.L. Wu, and M. Marek-Sadowska, "An Efficient Router for 2-D Field Programmable Gate Arrays", *Proc. of EDAC,* pp. 412-416, 1994.

[10]. Y.L. Wu, and D. Chang, "On NP-completeness of regular 2-D FPGA Routing Architectures and a Novel Solution", *Proceedings of ICCAD,* pp. 362-366, 1994.

[11]. A. Hashimoto and J. Stevens, "Wire Routing by Optimizing Channel Assignment with Large Apertures", *Proc. of the 8th Design Automation Workshop,* pp. 155-169, 1980.

[12]. Y. Sun, T.C. Wang, C.K.Wong, and C.L.Liu, "Routing for Symmetric FPGAs and FPICs", *Proc.of ICCAD,* pp. 486-490, 1993

[13]. Y, Sun, and C.L.Liu, "Routing in a New 2-D FPGA/FPIC Routing Architecture", *Proc.of DAC,* pp. 171-176, 1994

[14]. T. Ohtsuki at al, "One-Dimensional Logic Gate Assignment and Interval Graphs", *IEEE Trans. on Circuits and Systems,* Vol. CAS-26, No. 9, pp. 675-684, 1979.

Table 1. Coupling effect: by coupling two orthigonal algorithms (GG_S@GG_M), the decaying effect is very much reduced.

	seg_use% \ domain	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	11th	12th	13th	14th
alu4	GG_S	72.8	68.5	68.2	65.4	63.8	60.7	54.8	53.9	49.3	41.0	22.8	13.3	3.8	
	GG_M	65.9	67.5	60.7	58.7	59.7	56.1	50.0	56.6	46.23	44.1	38.0	24.8	5.9	1.0
	GG_S@GG_M	72.8	68.5	68.2	65.4	63.8	60.7	54.8	*50.3	*49.5	*44.1	*32.8	*9.0		
too-large	GG_S	65.7	64.2	67.5	65.2	60.6	55.5	50.4	52.7	41.4	38.1	13.0	5.1	1.0	
	GG_M	64.5	59.3	62.7	59.6	54.7	51.9	50.6	46.3	41.7	35.0	34.0	6.9		
	GG_S@GG_M	65.7	64.2	67.5	65.2	60.6	53.5	50.4	*48.6	*45.8	*37.1	*17.1			
example2	GG_S	73.9	69.7	66.8	63.6	63.9	56.8	58.4	54.8	43.6	47.1	36.1	19.7	8.1	4.5
	GG_M	68.7	66.8	65.2	60.3	55.8	65.2	53.9	52.9	44.2	49.4	39.4	24.2	4.8	
	GG_S@GG_M	73.9	69.7	66.8	63.6	63.9	56.8	58.4	*53.2	*51.3	*43.2	*35.2	*31.6		

*dd.d: This is the segment usage (%) in the routing domains when GG_S is coupled with GG_M

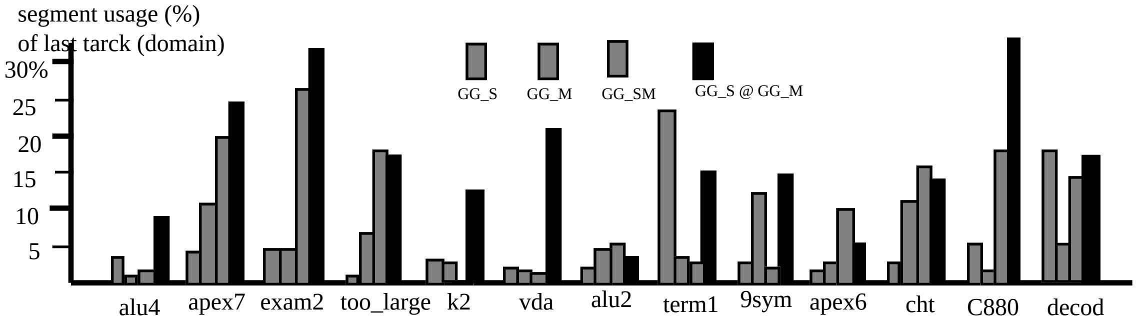

Fig. 7 Last track segment usage (%) of different schemes (Most schems, except GG_S@GG_M, exhibit sharp decaying effect and have very low segment usage of the last track domain)

Table 2. Comparisons between a conventional maze router, 2-step routers (CGE, SEGA) and our two greedy-grow routers and their variant couplings

odd domains even domains

Circuits	# of 4-LUTs	# of 2-pin nets	Maze W	CGE (SEGA) W	GG_S		GG_M		GG_SM		GG_S/GG_M		*GG_S @ GG_M	
					W	av.seg (%)	W	av.seg (%)	W	av.seg (%)	W	av.seg (%)	W	av.seg (%)
alu4	255	851	20	15	13	0.49	14	-0.18	14	8.75	13	4.59	12	0.75
apex7	80	300	15	13	11	0.26	11	-4.47	11	8.00	11	1.12	10	0.60
example2	120	444	21	18(17)	14	0.29	13	-2.13	12	3.11	13	1.31	12	0.39
too_large	156	519	17	13(12)	13	2.53	12	0.18	12	7.32	12	4.47	11	2.03
k2	360	1256	27	19 (17)	17	2.53	17	0.69	20	10.2	16	3.60	16	1.86
*vda	210	722	19	14 (13)	13	0.45	14	1.67	14	9.43	14	3.76	12	-0.09
alu2	143	511	17	12 (11)	11	-1.68	11	-3.66	11	2.74	12	1.42	11	-1.68
term1	56	202	13	10	9	-3.97	11	-1.23	11	2.87	10	1.50	9	-3.42
9symml	72	259	12	10	11	-7.60	9	-3.52	11	-0.23	9	-2.50	9	-7.49
apex6	255	982	unknown	32	17	1.75	18	-0.87	17	6.34	16	5.49	16	0.48
cht	48	202	unknown	16	12	-1.53	12	-1.53	11	2.30	11	0.13	11	-1.41
C880	120	427	unknown	12	12	0.64	13	0.00	12	6.86	12	1.33	11	-1.17
decod	20	88	unknown	9	6	-12.6	6	-16.5	6	-10.4	6	-11.3	6	-12.6
Total tracks			(161) (+59%)	187 [+28%]	159		161		162		155		(101) 146	

(ddd): summation of Ws for circuits with known maze router results

*: GG_M switched in from the 8th domain for every circuit except vda (from 12th)