Programming Assignment #3

B11901029 電機三 李致頡

Algorithm

My algorithm is basically the same as lines 7-17 in the multilevel global placement algorithm proposed in NTUplace3[1], but with some modifications:

1. **Wirelength model**: I use Stable Weighted-Average (WA) model proposed in [2] to approximate HPWL. For the hyperparameter γ , I follow [2] to set its initial value as $\gamma_o = 0.05 \times$ chip width. However, after the modules have already spread to a certain level, I decrease γ to $0.3\gamma_o$ to approximate HPWL better.

2. **Cache for gradient computation**: the formula of stable WA model is given by

$$W(\mathbf{x},\mathbf{y}) = \sum_{e\in E}\left[\frac{\sum_{v_i\in e} x_i \exp(\frac{x_i - x_{\max}}{\gamma})}{\sum_{v_i\in e}\exp(\frac{x_i - x_{\max}}{\gamma})} - \frac{\sum_{v_i\in e} x_i \exp(\frac{-x_i + x_{\min}}{\gamma})}{\sum_{v_i\in e}\exp(\frac{-x_i + x_{\min}}{\gamma})}\right.$$
$$\left. + \frac{\sum_{v_i\in e} y_i \exp(\frac{y_i - y_{\max}}{\gamma})}{\sum_{v_i\in e}\exp(\frac{y_i - y_{\max}}{\gamma})} - \frac{\sum_{v_i\in e} y_i \exp(\frac{-y_i + y_{\min}}{\gamma})}{\sum_{v_i\in e}\exp(\frac{-y_i + y_{\min}}{\gamma})}\right]$$

For a single term $W_1(\mathbf{x},\mathbf{y}) = \frac{\sum_{v_i\in e} x_i \exp(\frac{x_i - x_{\max}}{\gamma})}{\sum_{v_i\in e}\exp(\frac{x_i - x_{\max}}{\gamma})}$, we need to calculate its partial derivative with respect to the x-coordinate x_k of a particular module m_k. First define the pin set $V_k = \{\text{pin } v \mid (v \in e) \cap (v \text{ is on module } m_k)\}$, then the partial derivative is given by:

$$\frac{\partial W_1(\mathbf{x},\mathbf{y})}{\partial x_k} = \frac{\sum_{v_j\in V_k}\left[\frac{1}{\gamma} x_j \exp\left(\frac{x_j - x_{\max}}{\gamma}\right) + \exp\left(\frac{x_j - x_{\max}}{\gamma}\right)\right]}{\sum_{v_i\in e}\exp\left(\frac{x_i - x_{\max}}{\gamma}\right)}$$
$$- \frac{\left[\sum_{v_j\in V_k}\exp\left(\frac{x_j - x_{\max}}{\gamma}\right)\right]\left[\sum_{v_i\in e} x_i \exp(\frac{x_i - x_{\max}}{\gamma})\right]}{\gamma\left[\sum_{v_i\in e}\exp\left(\frac{x_i - x_{\max}}{\gamma}\right)\right]^2}$$

Notice that there are terms $\sum_{v_i\in e}\exp\left(\frac{x_i - x_{\max}}{\gamma}\right), \sum_{v_i\in e} x_i \exp\left(\frac{x_i - x_{\max}}{\gamma}\right)$, which exists in $W(\mathbf{x},\mathbf{y})$, so we build a table to memorize $\sum_{v_i\in e} x_i \exp\left(\frac{x_i - x_{\max}}{\gamma}\right)$, $\sum_{v_i\in e}\exp\left(\frac{x_i - x_{\max}}{\gamma}\right), \sum_{v_i\in e} x_i \exp\left(\frac{-x_i + x_{\min}}{\gamma}\right), \sum_{v_i\in e}\exp(\frac{-x_i + x_{\min}}{\gamma})$ for each net e when computing $W(\mathbf{x},\mathbf{y})$, avoiding iterating all pins on net e and computing the costly exponential function again. And same for the y-coordinate.

3. **Termination condition of CG solver**: the original CG solver in [1] terminate when $f(x_k) > f(x_{k-1})$. I find that the improvements $|f(x_k) - f(x_{k-1})|$ are very tiny for many steps before $f(x_k) - f(x_{k-1}) > 0$ eventually, which waste the runtime a lot but get even worse wirelength. Therefore, I set a certain threshold for the improvement: if the improvement ratio $\frac{f(x_{k-1})-f(x_k)}{f(x_{k-1})} < 0.002$, then the CG solver stops, modify the cost function by multiplying λ, and start the next iteration.

4. **Target Density**: the value of $t_{density}$ is not specified in the experimental result of ICCAD'04 IBM mixed-size benchmark suite in [1]. For this benchmark suite, the utility rates are all around 80%. However, it's not necessary to use the utility rate as the target density, since our goal is only to minimize wirelength while making the placement legalizable, rather than completely spread the modules. Therefore, I use a relatively high $t_{density} = \max(0.9,\ \text{utility rate})$ for most cases.

5. **Parallelization**: To accelerate the computation of cost functions and their gradients, and exploit the independence of net-wise and module-wise computations, we parallelized several performance-critical loops using OpenMP. While this significantly reduced runtime, the results may exhibit slight numerical differences due to non-deterministic floating-point accumulation in parallel reductions.

Result

Run my program on EDA Workstation edaU12.

	w/o parallelization			w/ parallelization		
Circuit	HPWL	Runtime	Score	HPWL	Runtime	Score
ibm01	51258709	9 sec	8.91168	51439499	3 sec	9.45291
ibm05	9591659	15 sec	8.80141	9589783	6 sec	9.08757

Reference

[1] T. -C. Chen, Z. -W. Jiang, T. -C. Hsu, H. -C. Chen and Y. -W. Chang, "NTUplace3: An Analytical Placer for Large-Scale Mixed-Size Designs With Preplaced Blocks and Density Constraints," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 27, no. 7, pp. 1228-1240, July 2008, doi: 10.1109/TCAD.2008.923063.

[2] M. -K. Hsu, V. Balabanov and Y. -W. Chang, "TSV-Aware Analytical Placement for 3-D IC Designs Based on a Novel Weighted-Average Wirelength Model," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 32, no. 4, pp. 497-509, April 2013, doi: 10.1109/TCAD.2012.2226584.