

求解 VLSI 布线问题的离散粒子群优化算法

刘耿耿¹ 王小溪¹ 陈国龙^{1,2} 郭文忠¹ 王少铃¹

(福州大学数学与计算机科学学院 福州 350108)¹

(福州大学离散数学及其应用教育部重点实验室 福州 350003)²

摘要 最小矩形 Steiner 树问题是 VLSI 布线的一个关键问题,且是一个典型的 NP 完全问题。为了有效地解决 VLSI 布线中考虑障碍物的最小矩形 Steiner 树问题,提出了一种改进的离散粒子群优化算法。考虑到存在障碍物,设计了一个基于惩罚的适应度函数。引入了遗传算法的变异和交叉算子,增加了种群的多样性并适当地扩展了粒子的寻优范围。实验结果表明,算法是有效的,实现简单,且相对遗传算法能更有效迅速地收敛。

关键词 超大规模集成电路,布线,粒子群优化,最小矩形 Steiner 树,障碍物

中图分类号 TP18 文献标识码 A

Discrete Particle Swarm Optimization Algorithm for the Routing of VLSI Circuit

LIU Geng-geng¹ WANG Xiao-xi¹ CHEN Guo-long^{1,2} GUO Wen-zhong¹ WANG Shao-ling¹

(College of Mathematics and Computer Sciences, Fuzhou University, Fuzhou 350108, China)¹

(Key Laboratory of Discrete Mathematics with Applications of Ministry of Education, Fuzhou University, Fuzhou 350003, China)²

Abstract Rectilinear Steiner Minimal Tree is one of the key problems in the routing of Very Large Scale Integration and a typical NP-complete problem. To solve the rectilinear Steiner minimal tree with rectangular obstacles(RSMTRO) problem effectively, an improved discrete particle swarm optimization(IDPSO) algorithm was proposed. Considering existence of obstacles, the penalty-based fitness function was designed. The principles of mutation and crossover operator in genetic algorithm were incorporated into the proposed PSO algorithm to achieve better diversity, and the scope of the particle optimization was appropriately expanded. Simulation results show that IDPSO algorithm can efficiently provide RSMTRO solution with good quality and converges more efficiently and rapidly than genetic algorithm.

Keywords Very large scale integration, Routing, Particle swarm optimization, Rectilinear steiner minimal tree, Obstacles

1 引言

布线问题是超大规模集成电路(Very Large Scale Integration, VLSI) 布图设计中的一个重要环节。在 VLSI 布线问题中,对连接树的要求为:其边平行于芯片的边且不能穿越芯片的功能区域,达到期望目标函数最优,即连接树的总长度最短。布线问题中多端线网的最佳连接是构造最小矩形 Steiner 树,该问题已被证明是一个 NP 完全问题^[1]。学者对此问题进行了大量的研究,并提出了一些启发式算法,大多数都是基于最小生成树的启发式算法。文献[2]提出了一种基于改进的 Prim 启发算法寻找最小矩形 Steiner 树的方法;文献[3]提出了一个关于最小矩形 Steiner 树的混合遗传算法,表明遗传算法具有较好的效果。在考虑障碍物的情况下,构造最小矩形 Steiner 树变得更加复杂。对于构造考虑障碍物的最小矩形 Steiner 树(Rectilinear Steiner Minimal Tree with

Rectangular Obstacles, RSMTRO)问题, Wey Chin-Long^[4]给出了一种基于路径构造和路径调度的求解 RSMTRO 问题的启发式算法。Rita 和 Bryant^[5]针对求解 RSMTRO 问题提出了一种基于最小生成树的有效遗传算法,表明了智能优化算法在该类问题中具有较好的应用前景。

粒子群优化(Particle Swarm Optimization, PSO)算法是 1995 年由 Eberhart 博士和 Kennedy 博士提出的,是基于群智能优化的算法^[6]。PSO 算法相对于其他进化算法而言,其最吸引人的特征是具有实现简单和更强的全局优化能力。大量实验结果表明了 PSO 算法能够解决遗传算法所能解决的各种优化问题,也显示出 PSO 算法确实是有力的优化工具,且具有强大的生命力^[7]。本文提出一种改进的离散 PSO(Improved Discrete Particle Swarm Optimization, IDPSO)算法来解决 RSMTRO 问题。在 IDPSO 算法中,粒子编码采用 Prüfer 数编码机制和 01 序列编码结合起来表示一棵矩形 Stei-

到稿日期:2009-11-18 返修日期:2010-02-01 本文受国家 973 重点基础研究发展规划项目(No. 2006CB805904),国家自然科学基金项目(No. 10871221),福建省科技创新平台计划项目(2009J1007),福州大学大学生科研训练计划资助。

刘耿耿(1988—),男,硕士生,主要研究方向为计算智能及其应用,E-mail: tianyishui880126@126.com;王小溪(1986—),女,硕士生,主要研究方向为计算智能及其应用;陈国龙(1965—),男,教授,博士生导师,主要研究方向为人工智能、网络信息安全等;郭文忠(1979—),男,博士生,讲师,主要研究方向为计算智能、计算机网络等。

ner 树,其中 Prufer 数编码构成矩形 Steiner 树的基础生成树,01 序列编码 Steiner 点的选择。考虑到存在障碍物,IDP-SO 算法设计了一个基于惩罚的适应度函数,引入了遗传算法的变异和交叉算子,增加了种群的多样性和适当地扩展了粒子的寻优范围。对几组布线模型实例进行了仿真实验,实验结果表明 IDPSO 算法是有效的,实现简单,且相对遗传算法能更有效迅速地收敛。

2 问题模型的建立和分析

2.1 RSMTRO 问题模型

设端点集合 $P = \{P_1, P_2, P_3, \dots, P_n\}$, 作为 n 个芯片引脚的集合,每个引脚 P_i 对应一个点坐标 (x_i, y_i) 。矩形集合 $R = \{R_1, R_2, R_3, \dots, R_m\}$, 作为 m 个矩形障碍物的集合,每个矩形障碍物 R_i 可用其相应的左下顶点和右上顶点表示。图 1 所示的模型是 7 个芯片引脚、3 个矩形障碍物的 RSMTRO 问题模型。引脚 1 对应的点坐标为 $(12, 1)$; 矩形障碍物 R_1 可表示为 $(3, 7, 5, 14)$, 其中 $(3, 7)$ 表示 R_1 左下顶点的坐标, $(5, 14)$ 表示 R_1 右上顶点的坐标。在原来端点集合 P 的基础上,引入一些特殊的点(Steiner 点),连接所有的芯片引脚,连接树的每条边都是直角矩形边且边不穿越任意一个矩形障碍物 R_i , 达到期望目标函数值最优(连接树的总长度值最小),称为 RSMTRO 问题模型的求解。

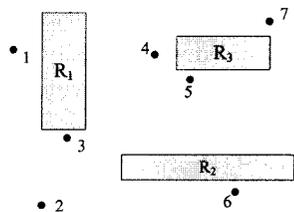


图 1 一个 RSMTRO 问题模型示例(7 个布线端点、3 个矩形障碍物)

定义 1 一棵 Steiner 树中,若该 Steiner 树的每条边均是直角矩形边,则该 Steiner 树称为矩形 Steiner 树(Rectilinear Steiner Trees, RST)。

定义 2(0 选择) 如图 2(1)所示, (x_1, y_1) 和 (x_2, y_2) 为边 L 的两个端点,即端点 1 和端点 2,其中 $x_1 < x_2$ 。边 L 对应的 Steiner 点的选择如图 2(2)所示,从端点 1 先引竖直边再引水平边至端点 2,则称作 0 选择。图 2(2)中的 S 点即为 0 选择 Steiner 点。

定义 3(1 选择) 如图 2(1)所示, (x_1, y_1) 和 (x_2, y_2) 为边 L 的两个端点,即端点 1 和端点 2,其中 $x_1 < x_2$ 。边 L 对应的 Steiner 点的选择若如图 2(3)所示,从端点 1 先引水平边再引竖直边至端点 2,则称作 1 选择,图 2(3)中的 S 点即为 1 选择 Steiner 点。

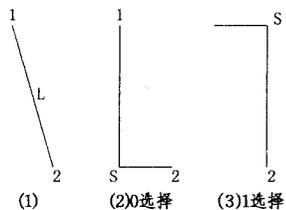


图 2 Steiner 点的两种选择情况

2.2 RSMTRO 问题分析

定理 1^[8] 关于布线端点集合 P , 必定存在一棵最小矩

形 Steiner 树,其 Steiner 点集合 S 是 Hanna 网格交点集合 U 的子集。

在 RSMTRO 问题所给定待连接的布线端点集合中, 通过从这些端点分别各自引一条水平线和竖直线, 这些线的交叉点就形成 Hanan 点集合^[8]。从 Hanan 点集合中按照一定规则选取的一部分点作为 Steiner 点集合, 这些 Steiner 点与普通的生成树一起构成了 RST。考虑到芯片上存在着布线不能穿越的功能模块, 这样就衍化出了本文上面提出的考虑障碍物的最小矩形 Steiner 树问题, 即 RSMTRO 问题。RSMTRO 问题与不考虑障碍物的 RST 问题相比, 计算更加复杂。一般来说, 障碍物可以是任意形状的。但在 VLSI 布线区域中, 存在着连接树的边不能穿越的功能模块, 这些模块的边要求平行于芯片的边。

定义 4(连通模型) 在布线模型中, 障碍物的存在可能会阻止部分 Hanan 点被选取成为 Steiner 点。但是每个布线端点都可以与其他布线端点中至少一个端点通过一次 0 选择或 1 选择连接, 如图 3(1)所示的模型即为连通。

定义 5(非连通模型) 在布线模型中, 至少存在一个布线端点只能通过两次或两次以上的 0 选择或 1 选择跟其他布线端点连接, 这样引入的 Steiner 点并不是基于 Hanan 点选取的 Steiner 点, 如图 3(2)所示的模型即为非连通。

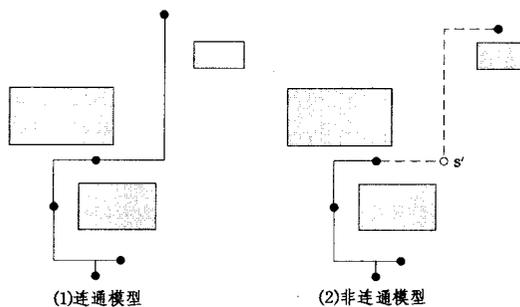


图 3 RSMTRO 问题的两类情况

3 求解 RSMTRO 问题的有效离散 PSO 算法

3.1 基本 PSO 算法

基本 PSO 算法是基于群体和适应度的优化算法, 将每个个体看作 D 维搜索空间中的一个没有体积的微粒, 有自己的位置向量和速度向量, 在搜索空间中以一定的速度飞行。该速度是根据粒子本身的历史最优位置和该种群的全局最优位置来计算的。第 i 个粒子表示为 $X_i = (X_{i1}, X_{i2}, \dots, X_{id})$, 它经历过的最好位置记为 $P_i = (P_{i1}, P_{i2}, \dots, P_{id})$, 也称为 P_{pb} 。在种群中所有粒子经历过的最好位置的索引号用符号 P_g 表示, 也称为 P_{gb} 。粒子 i 的速度用 $V_i = (V_{i1}, V_{i2}, \dots, V_{id})$ 表示。对每一代根据如下方程变化:

$$V_i^{t+1} = \omega \times V_i^t + c_1 r_1 (P_{pb} - X_i^t) + c_2 r_2 (P_{gb} - X_i^t) \quad (1)$$

$$X_i^{t+1} = X_i^t + V_i^t \quad (2)$$

式中, ω 为惯性权重, c_1 和 c_2 为加速因子, r_1 和 r_2 为两个在 $[0, 1]$ 范围内变化的随机数。

3.2 求解 RSMTRO 问题的改进的离散 PSO 算法

由于 RSMTRO 问题是一个离散的优化问题, 而基本 PSO 算法是针对连续的优化问题设计的, 无法直接用来解决 RSMTRO 这类离散问题的求解, 因此本文设计了改进的离散型 PSO 算法来求解 RSMTRO 这一类离散问题。自从粒子群

优化算法提出之后,很多学者尝试进行构造求解离散问题的 PSO 算法。Kennedy 和 Eberhart 提出一种离散二进制 PSO 算法^[9], Clerc 提出一种求解 TSP 问题的离散 PSO 算法^[10]。而在前期工作中,文献[11]提出了求解异构机群数据分配问题的改进离散 PSO 算法;文献[12]针对 VLSI 线长估计中的多目标最小生成树问题,构造了一种有效的离散 PSO 算法。为此,本文在文献[12]的研究基础上,结合 Prufer 数和 01 序列编码一棵 RST 树,建立了基于惩罚的适应度函数,继而给出了一种用于求解 RSMTRO 问题的 IDPSO 算法。

3.2.1 粒子的编码

定义 6(Prufer 编码)^[13] 给定一棵树 T 是一棵有 n 个顶点的标号树,其顶点编号为 $\{1, 2, \dots, n\}$ 。根据 Cayley 定理,共有 n^{n-2} 棵不同的有 n 个顶点的标号树。Prufer 给出了这 n^{n-2} 棵标号树与标号序列 c_1, c_2, \dots, c_{n-2} 之间的一一对应关系,其中 $1 \leq c_i \leq n, 1 \leq i \leq n-2$ 。序列 c_1, c_2, \dots, c_{n-2} 就称为标号树的 Prufer 编码。

由于篇幅所限, Prufer 编码过程和解码过程详见文献[13]。

假设 1 RMSTRO 问题模型中障碍物的存在可能会阻止部分 Hanan 点被选取成为 Steiner 点,但是每个布线端点都可以与至少一个其他布线端点通过一次 0 选择或 1 选择连接。

假设 2 RMSTRO 问题模型的障碍物形状是矩形的。

IDPSO 算法解决的 RMSTRO 问题是要求该问题模型的布线端点满足假设 1 与假设 2 的条件。假设 1 就是要求 RMSTRO 问题模型是属于连通模型。RSMTRO 问题的布线模型中障碍物如果不是矩形的,如图 4(1)所示的 Rect1, 本文可以将 Rect1 看作是由图 4(2)中所示的 Rect2 和 Rect3 两个矩形组成。因此,无论多么复杂的图形,都可以将它转变成多个矩形的合成,假设 2 容易得到满足。

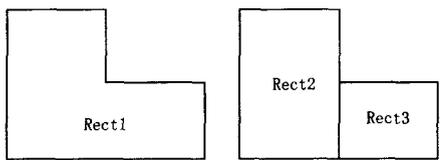


图 4 一个复杂形状的障碍物分解

在满足假设 1 和假设 2 的情况下,本文采用 Prufer 数的编码机制编码构成 RST 树的基础生成树。如图 5(1)所示,它是构成图 5(2)中的 RST 树的基础生成树,而 RST 树还包括对 Steiner 点的选择,所以本文引入 01 序列编码 RST 树的 Steiner 点的选择,其中 01 序列的 0 代表 0 选择,1 代表 1 选择。Prufer 数的编码机制,再结合 RST 树的 Steiner 点的选择,共同编码一棵 RST 树,如图 5(2)所示。

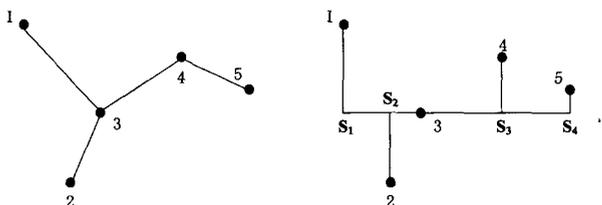


图 5 RST 树与其基础生成树

(1)RST 树的基础生成树 (2)RST 树(其中 s_1, s_2, s_3, s_4 是 Steiner 点)

基于以上的 Prufer 数的编码机制和 Steiner 点的选择规则,本文结合两者来编码一棵确定的 RST 树。本文将种群示成 $(2 * n * popNum)$ 的三维矩阵,其中 n 为顶点数, $popNum$ 为种群规模。粒子矩阵的第一行的第 1 到 $(n-2)$ 位为该模型所对应 RST 的基础树的 Prufer 数编码,第 $n-1$ 位用来统计 RST 的直角边穿越障碍物的次数,第二行的第 1 到 $(n-1)$ 位为该模型所对应 RST 的斯坦纳点选择序列,第 n 列的元素表示该粒子的适应度函数值。图 6 给定的 RST 树可表示为种群中的一个粒子,它的编码可以表示成:

3 4 2 3 4 0 40
0 0 1 1 1 0 40

其中 RST 树的基础生成树所对应的 Prufer 编码序列为 3 4 2 3 4, 其 Steiner 点的选择序列为 0 0 1 1 1 0, 最后一列用来存储该粒子的适应度函数值。本文中种群的初始化是随机产生 $popNum$ 个上述形式的粒子,作为 IDPSO 算法迭代的初始种群。

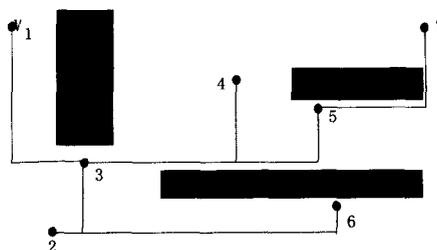


图 6 一棵矩形斯坦纳树(7 个布线端点、3 个障碍物)

3.2.2 粒子的适应度函数

定义 7(惩罚函数)^[14] $P(X, Q) = N(X) + Q * S(X)$, 其中 $P(X, Q)$ 称为惩罚函数, $Q * S(X)$ 为惩罚项。其中 Q 为惩罚因子,是极限为 ∞ 的数列。

定义 8^[15] 一棵 Steiner 树的长度为该 Steiner 树所有边长度的和,即:

$$L(T_X) = \sum_{e_i \in T_X} l(e_i) \quad (3)$$

式中, $l(e_i)$ 为边 e_i 的长度。有时,也把 Steiner 的长度称为费用。

惩罚函数法是一种应用广泛而且行之有效的优化方法,它的基本思想是对有约束的最优化问题的求解过程中企图违反约束的那些迭代点,给予目标函数以很大的惩罚,以转化为无约束问题。本文在解决 RSMTRO 问题中考虑到障碍物存在的约束,故采用基于惩罚的适应度函数来评价可行解的优劣,把 RSMTRO 问题转换成无约束问题。即本算法的适应度函数为:

$$F(X) = N(X) + Q * S(X) = 2^{cf} * L(T_X) \quad (4)$$

式中, $N(X) = L(T_X)$, $S(X) = L(T_X)$, $Q = 2^{cf} - 1$, cf 为解码后的 RST 树中直角矩形边穿越障碍物的次数。

采用基于惩罚的适应度评价函数,对存在违反约束的粒子,即粒子所对应的 RST 树中有穿越障碍物的直角矩形边,并没有在下次种群迭代前淘汰掉违反约束的粒子,保留了粒子可能携带的部分最优信息,让像这样违反约束的粒子在以后的迭代中有机会将其可能携带的部分最优信息表现出来。这样,基于惩罚的适应度函数一方面评价了粒子的优劣性,另一方面又能让粒子携带的部分最优信息得以保存和让种群保持较高的多样性。

3.2.3 粒子的更新公式

RMSTRO 问题是离散型的最优化问题,而传统的 PSO 算法的粒子更新操作是针对连续型的最优化问题,已经不再适合解决 RMSTRO 问题,因此 IDPSO 改进了粒子更新操作。首先,粒子先前速度的保持使用遗传算法中的变异算子替代,即:

$$W_i = N_1(X_i^{i-1}, w) = \begin{cases} M(X_i^{i-1}), & rand_1 < w \\ X_i^{i-1}, & \text{其他} \end{cases} \quad (5)$$

式中, X_i^{i-1} 表示第 i 个粒子在第 $i-1$ 代的可行解; M 函数同遗传算法中的变异算子,即满足 $r_1 < w$ 时,将在粒子矩阵第一行的 Prufer 数编码序列随机产生两个位置,交换对应的作业编码。但对粒子矩阵第二行的 01 编码的操作方式不一样:随机产生一个位置,将该位置的数值加 1 模 2。

其次,粒子的自我认知及社会认知部分分别使用遗传算法中的交叉算子代替,即:

$$S_i = N_2(W_i, c_1) = \begin{cases} C_p(W_i), & rand_2 < c_1 \\ W_i, & \text{其他} \end{cases} \quad (6)$$

$$X_i = N_3(S_i, c_2) = \begin{cases} C_g(S_i), & rand_3 < c_2 \\ S_i, & \text{其他} \end{cases} \quad (7)$$

式中, $C_p(W_i)$ 算子表示将变异后的粒子 W_i 与第 i 个粒子的历史最优解 P_{best} 进行交叉运算, $C_g(S_i)$ 算子表示将 S_i 与种群的全局最优解 P_{gbest} 进行交叉运算。交叉算子:随机产生的一个数,若大于加速因子 c_1 或 c_2 ,将随机产生两个位置,将这两个位置之间的序列分别和粒子的历史最优值矩阵或是种群全局最优值矩阵这两个位置之间的序列交换。

3.2.4 本文算法步骤

本文提出的 IDPSO 算法步骤如下。

Step1 初始化各类参数和随机生成初始种群。

Step2 评价各个粒子的适应度函数值,初始化粒子的历史最优解和全局最优解。

Step3 根据惯性权重 w 值对每个粒子进行变异操作,根据加速因子 c_1, c_2 对每个粒子进行相应的交叉操作。

Step4 重新评价各粒子的适应度函数值,更新各个粒子的历史最优解。

Step5 更新种群的全局最优解。

Step6 若满足终止条件,则循环结束;若不满足则返回 Step3。

4 实验结果与分析

4.1 IDPSO 算法的参数设定

本实验中,惯性权重 w 初始值 w_{start} 为 0.9,结束值 w_{end} 为 0.4;加速因子 c_1 初始值 c_{1_start} 为 0.9,结束值 c_{1_end} 为 0.6;加速因子 c_2 初始值 c_{2_start} 为 0.4,结束值 c_{2_end} 为 0.9。

w, c_1, c_2 均采用 Shi^[16] 提出的线性递减的思想,在 IDPSO 算法的循环主体中,参数 w, c_1, c_2 在每一次迭代中分别依式(8)~式(10)变化:

$$w = w_{start} - \frac{w_{start} - w_{end}}{evaluations} \times eval \quad (8)$$

$$c_1 = c_{1_start} - \frac{c_{1_start} - c_{1_end}}{evaluations} \times eval \quad (9)$$

$$c_2 = c_{2_start} - \frac{c_{2_start} - c_{2_end}}{evaluations} \times eval \quad (10)$$

式中, $eval$ 是当前迭代次数, $evaluations$ 是总的迭代次数。种群的大小 $popSize$ 和算法的迭代次数 $evaluations$ 的设定与 RSMTRO 问题的布线端点和功能障碍物的规模有关。

4.2 IDPSO 算法与遗传算法的最优值进化曲线的比较和分析

本文给定的一个布线测试实例是 13 个待连接的布线端点与 4 个矩形功能障碍物。经过 IDPSO 算法和遗传算法对该实例分别进行试验,得到了这两种算法的最优值进化曲线,如图 7 所示。在本测试实例中, IDPSO 算法在迭代 20 多代后基本收敛到一个近似最优值 32,即布线模型中矩形 Steiner 树的长度,而遗传算法要迭代 160 多代后才收敛到一个近似最优值 34,可以看出本文的 IDPSO 算法相对于遗传算法,能在较少的迭代次数后收敛到一个近似最优值,收敛速度较快,且算法的操作简单,不用进行复杂的迭代操作。

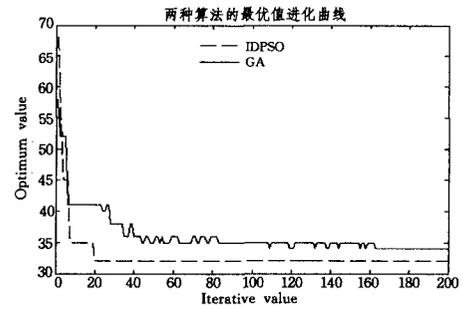


图 7 两种算法的最优进化曲线比较

4.3 两种算法平均适应度函数值的比较和分析

本文给定一个 7 个端点、3 个矩形功能障碍物的布线测试实例,采用 IDPSO 算法进行随机的 12 次测试,得到的适应度函数值分别为 41, 39, 39, 41, 41, 42, 41, 46, 41, 43, 39, 41, 平均值为 41.17, 标准差为 1.86。其中比较好的最优值 39 出现了 3 次,而比较不理想的结果(43, 46)出现了两次。而采用遗传算法进行随机的 12 次测试,得到的适应度函数值分别为 43, 50, 43, 46, 45, 43, 41, 41, 50, 41, 39, 43, 平均值为 43.75, 标准差为 3.3。其中比较好的最优值 39 出现了一次,比较不理想的结果(45, 46, 50)出现了 4 次。

IDPSO 相对于 GA 来说,平均得到比较好的布线方案的次数比较多,得到好的结果比较稳定,出现比较不理想的次数比较少。在布线端点和障碍物数较多的情况下, IDPSO 算法也能得到比较好的结果。图 8 给出了一个布线端点数为 26、障碍物数为 21 的 RSMTRO 问题的近似最优的解决方案,而遗传算法就很难取得这样的优化结果。总体来说, IDPSO 在解决 RSMTRO 问题上比遗传算法更具可行性和有效性,可取得相对遗传算法更好的布线结果。

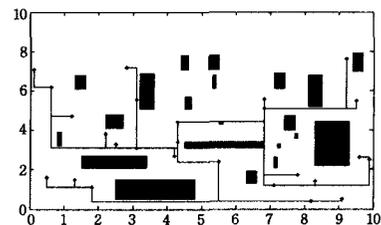


图 8 布线端点数为 26、障碍物数为 21 的最小矩形 Steiner 树的构造

结束语 PSO算法的搜索机制是随机的,通过迭代计算来找到近似最优值或是问题的近似最优解决方案,无论适应度函数是不是连续的或是可导的,都可以应用 PSO 来解决。因此,PSO 的应用范围就更广泛,更有可行性。本文提出的 IDPSO 是改进的离散型 PSO 算法,在 IDPSO 算法中融入了遗传算法的部分操作算子,包括变异算子和交叉算子。普通的 PSO 算法一般以一定的概率按照式(1)、式(2)更新粒子的位置和速度,可以比较迅速地得到次优适应度函数值,但是有时候会因为忽略的粒子有较好的优化信息,从而陷入局部最优。IDPSO 通过粒子群优化算法本身的更新规则代替遗传算法中的选择规则,在每次的迭代过程中都以一定的概率对每个粒子进行交叉、变异操作,尽可能地保留种群中各个粒子的优化信息,尽可能避免迅速陷入局部最优。

本文采用的 IDPSO 算法通过多组实例的测试得到了比较不错的布线方案;通过与遗传算法在最优值进化曲线和平均适应度函数值的比较,表明 IDPSO 具有更快的收敛速度、简单的操作,更具可行性和有效性。在下一步的工作中,将寻找一种能够表示更为广泛的矩形 Steiner 树的编码,用以表示非连通布线模型中的矩形 Steiner 树。

参考文献

- [1] Hashimoto A, et al. Wire Routing by Optimizing Channal Assignment Within Large Apertures[C]//Proc. of 8th IEEE/ACM Design Automation Workshop. 1971
 - [2] 邓爱姣,李强,张嘉为.改进的 Prim 启发式算法在 VLSI 布线中的应用[J].沈阳工业大学学报,2006,28(5):557-559
 - [3] 杨昌玲,严晓浪.基于树形编码的 MRST 混合遗传算法及其并行处理[J].微电子学,1999,29(2):89-95
 - [4] Wey Chin-Long. Efficient Rectilinear Steiner Tree Construction with Rectangular Obstacles[C]//Proc. 5th WSEAS International Conference on Circuits System Selectronics Control & Signal Processign. Dallas, USA, 2006;204-208
 - [5] Rita M H, Bryant A J. A Spanning-Tree-based Genetic Algorithm for Some Instances of the Rectilinear Steiner Problem with Obstacles[C]//Proceedings of the 2003 ACM Symposium on Applied Computing. 2003;725-729
 - [6] Eberhart R C, Kennedy J. A New Optimizer Using Particles Swarm Theory[C]//Proc. 6th International Symposium on Micro Machine and Human Science. Nagoya, Japan, 1995;39-43
 - [7] Clerc M, Kennedy J. The Particle Swarm-explosion, Stability, and Convergence in a Multidimensional Complex Space [J]. IEEE Transactions on Evolutionary Computation, 2002, 6(1): 58-73
 - [8] Hanan M. On steiner's problem with rectilinear distance[J]. SIAM Journal of Applied Mathematics, 1996, 14(2): 255-265
 - [9] Kennedy J, Eberhart R C. A Discrete Binary Version of the Particle Swarm Optimization Algorithm[C]//Proc. of the IEEE International Conference on Systems Man and Cybernetics. Orlando, USA, 1997, V:4104-4109
 - [10] Clerc M. Discrete Particle Swarm Optimization illustrated by the Traveling Salesman Problem[EB/OL]. <http://www.mauriceclerc.net>, 2000-02-29
 - [11] 郭文忠,陈国龙,夏添.异构机群下数据流自适应分配策略[J].计算机辅助设计与图形学学报,2009,21(8):1175-1181
 - [12] 郭文忠,陈国龙.一种求解多目标最小生成树问题的有效离散粒子群优化算法[J].模式识别与人工智能,2009,22(4):597-604
 - [13] 王晓东,吴英杰. Prufer 编解码的最优算法[J].小型微型计算机系统,2008,29(4):687-690
 - [14] 张会生,翁史烈,张小兵,等.基于内弹道改进型零维模型的装药优化仿真[J].弹道学报,2000,12(3):32-36
 - [15] 洪先龙,严晓浪,乔长阁.超大规模集成电路布图理论与算法[M].北京:科学出版社,1998
 - [16] Shi Y H, Eberhart R C. A Modified Particle Swarm Optimizer [C]//IEEE International Conference of Evolutionary Computation. Piscataway, NJ: IEEE, 1998; 69-73
-
- (上接第 164 页)
- [10] Kang Donghyun, Han Saeyoung, Yoo Seohee, et al. Prediction-based Dynamic Thread Pool Scheme for Efficient Resource Usage[C]//Proceedings of the 2008 IEEE 8th International Conference on Computer and Information Technology Workshops. 2008;159-164
 - [11] Goetz B. Thread Pools Help Achieve Optimum Resource Utilization [EB/OL]. http://www.900.ibm.com/developerWorks/en/java/j-jtpO730/index_eng.shtml, 2002-07
 - [12] 邵鸣年,张昕.一种分布式系统中线程池的设计与实现[J].计算机工程与设计,2005,26(1):7-11
 - [13] Mercury Interactive. LoadRunner controller user's guide windows version8.0 [EB/OL]. <http://www.mercury.com/us/products/performance-center/loadrunner/>
-
- (上接第 172 页)
- [10] Dimitris S, Antonios D, Timos S. Hierarchically compressed wavelet synopses[J]. The VLDB Journal, 2009, 18(1): 203-231
 - [11] Franky Kin-Pong C, Ada Wai-chee F, Clement Y. Haar Wavelets for Efficient Similarity Search of Time-series, With and Without Time Warping [J]. IEEE Trans. on Knowl. and Data Eng., 2003, 15(3): 686-705
 - [12] Popivanov I, Miller R J. Similarity Search Over Time-Series Data Using Wavelets[C]//Proceedings of the 18th International Conference on Data Engineering. IEEE Computer Society, 2002; 212-216
 - [13] Liabotis I, Theodoulidis B, Saraaee M. Improving Similarity Search in Time Series Using Wavelets[J]. International Journal of Data Warehousing and Mining, 2006, 2(2)
 - [14] Yingyi B, Lei C, Ada Wai-Chee F, et al. Efficient anomaly monitoring over moving object trajectory streams[C]//Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Paris, France, ACM, 2009; 159-168
 - [15] Hao-Ping H, Ming-Syan C. Efficient range-constrained similarity search on wavelet synopses over multiple streams[C]//Proceedings of the 15th ACM International Conference on Information and Knowledge Management. Arlington, Virginia, USA, ACM, 2006; 327-336