第 3 章

CHAPTER 3

Memetic 算法

Memetic 算法是进化算法(evolutionary computation)中的一种,它是在遗传算法的基础上加上局部优化搜索算子得到的,也被称为混合遗传算法(hybrid evolutionary algorithm)、文化算法(cultural algorithm)[1-2]、密母算法等。Memetic 算法继承了拉马克进化论思想。在进化论中,关于继承模型有两种:一是拉马克继承(Lamarckian inheritance)模型;二是班德文继承(Baldwinian inheritance)模型。这两种模型的主要区别在于个体在一生中所学到的技能是否应该遗传给下一代。拉马克继承模型认为后代应该继承其父代所学的技能,而班德文继承模型则不认可这样的继承[1]。

Memetic 算法可以看作在局部最优子空间上进行了特殊类型的遗传搜索,在遗传算法中加入了局部优化方法并将其应用于每个个体。局部优化方法还解决了因为交叉和变异产生的不在当前局部最优子空间中的解,使最终产生的解维持在当前子空间中。在这种混合方法中,遗传算法用于种群中的全局广度搜索,而启发式方法则用于进行染色体间的局部深度搜索。由于遗传算法和传统启发式方法具有互补性质,这种混合方法的性能通常比单独运行任意一种要好。

3.1 Memetic 算法发展历程

Memetic 算法由 Pablo Moscato 在 1989 年提出。遗传算法主要是模拟物种的进化过程,而 Memetic 算法主要是为了模仿文化进化过程[3-5]。其中,个体的局部搜索优化过程就是模拟人在成长过程中的学习过程;个体通过在其邻域搜索找到局部最优解来达到提高适应度函数的目的,正如人通过学习提高自身能力一样。

因 Memetic 算法优化性能远高于单纯的遗传算法,所以被广泛关注并应用到许多研究领域中。

改进的 Memetic 算法已在社区检测(community detection)问题中使用并取得了成功。如在文献[6]中,作者主要研究社区检测中分辨率受限的问题,并提出称为 Meme-Net 的 Memetic 算法来解决该问题。在文献[7]中,作者提出了一种可以有效解决社区检测问题的快速 Memetic 算法,该算法采用多级学习作为局部搜索算子,实验表明了算法的有效性。在文献[8]中,作者提出了一种快速的 Memetic 算法来解决社交网络的结构平衡问题。

Lacomme 等在 2001 年将 Memetic 算法应用到 CARR 问题的求解中,但它的不足在于

全局搜索能力不强[9]。2004 年 Lacomme 等又提出了一种更高效的模因演算算法 LMA[10]，通过对某些解执行重新进化和提高局部搜索概率，从而提高了解的收敛速度，但是其只增加了局部搜索能力，全局寻优能力依然不强。之后，唐柯等又在 LMA 的框架下引入了扩展步长的领域搜索(MAENS)方法，使算法可以跳出局部最优，并在测试集上获得良好的下界[11]。随着关于 LSCARP 的研讨越来越深入，2013 年，梅一等又将该算法与路由距离分组(RDG)方案[12]相结合提出了 RDG-MAENS 算法。在该算法中，首先采用 RDG 策略将路由问题中的任务集合分为多个小的任务集，再使用 MAENS 独立处理每个子任务集合。并且，Memetic 算法在多目标 CARP 的应用也越来越多，如 2011 年梅一等提出的 D-MAENS 算法[13]。

另外，Memetic 算法还经常被用来训练神经网络，Toby O. Hara 等[14]和 H. A. Abbass 等[15]分别用 Memetic 算法来训练神经网络，实验结果表明 Memetic 算法的寻优能力及搜索速度较其他传统方法更快。Tatiane R. Bonfim 等[16]不仅利用 Memetic 算法训练神经网络系统，还将其应用于并行机调度问题，两者都取得了很好的效果。

Memetic 算法因其较强的优化能力，使其还可用于解决 TSP、模糊系统控制、图像处理等优化问题中，在这里不再一一赘述，本书将在 3.3 节和 3.4 节详细描述 Memetic 算法在社区检测和限量弧路由问题上的应用。

3.2 Memetic 算法实现

3.2.1 Memetic 算法流程

可以认为 Memetic 算法是由遗传算法改进得到的，相比于遗传算法，Memetic 算法的重点就是多了局部搜索机制，所以该算法的算法流程图与遗传算法的算法流程有很多相似之处。一般地，Memetic 算法流程图可以由图 3.1 表示。如图 3.1 可知，Memetic 算法执行的具体步骤如下所述。

步骤 1：确定解的编码方法和算法执行中的所有参数；

步骤 2：初始化种群，得到父代群体；

步骤 3：执行交叉算子；

步骤 4：利用局部搜索算法对个体进行邻域搜索，更新种群所有个体；

步骤 5：执行变异算子；

步骤 6：再次利用局部搜索算法更新种群个体；

步骤 7：根据适应度函数计算种群内个体的适应度值；

步骤 8：执行选择算子，保留当前种群内的优秀个体(适应度值高的个体)，产生新种群；

步骤 9：判断是否满足终止条件，若满足终止条件则算法结束，输出优化结果；若不满足终止条件，则跳转到步骤 3 继续执行。

3.2.2 Memetic 算法改进

随着新问题的提出，问题规模的增大，人们对求解精度要求的提高等，在应用 Memetic 算法时，必须对算法本身进行改进，以适应新的问题和新的要求。Memetic 算法可改进的思路很多，可以从编码方式、种群初始化方法、交叉算子、变异算子、选择算子、局部搜索算子等

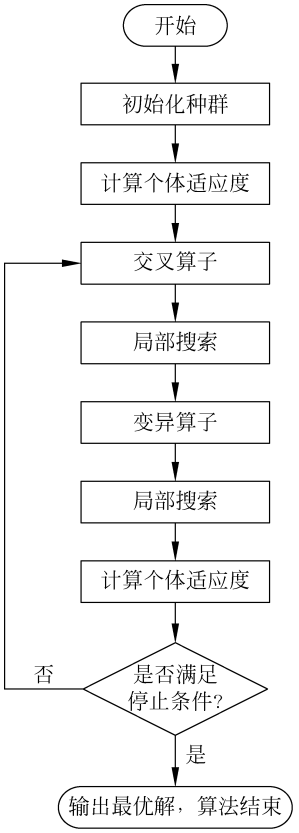

图 3.1 Memetic 算法流程图

多方面入手。

1. 编码方法

根据采用的符号，可以将编码方式分为二进制编码、实数编码和整数编码等；根据编码的长度可以将编码方式分为变长度编码和不可变长度编码。选择什么样的编码方式，需要根据特定问题进行选择，合适的编码方式有利于算法的执行，降低算法难度，提高算法效率等。

2. 种群初始化

种群初始化方式分为两种：在无先验知识时，采用完全随机的初始化方法；当有一定先验知识时，可以通过设定相关的机制，融入先验知识来生成初始种群。

3. 交叉算子

常用的交叉算子有单点交叉、两点交叉、多点交叉、均匀交叉等。对于特殊的问题也有特殊的交叉方式，如 Memetic 算法用于社区检测时，还可以使用双向交叉[17]。

4. 变异算子

变异算子可以在一定程度上提高局部搜索能力，增加种群多向性，但当问题较复杂时，就无法达到想要的效果。常用的变异方法包括单点变异、互换变异、插入变异等。

5. 选择算子

同遗传算法一样，Memetic 算法常用的选择操作有轮盘赌选择、竞标赛选择、排序选择

等，主要作用就是保留种群中的优秀个体到下一代，以实现优胜劣汰。

6. 局部搜索算子

从某些方面来看，可以把 Memetic 算法看成是遗传算法的改进方法，而局部搜索算子就是使 Memetic 算法不同于遗传算法的重要突破点，它在遗传算法中加入局部搜索机制，使传统的遗传算法具有全局和局部的搜索能力。常用的局部搜索机制有爬山算法、模拟退火算法、禁忌搜索算法等。针对不同问题，应选择不同的局部搜索方法，局部搜索效率越高，整个 Memetic 算法的效率也越高。

(1) 爬山算法。从当前解开始，搜索邻域计算适应度值，选出适应度值最高的个体作为山峰(最高点)，继续对新个体的邻域进行搜索，若当前最大适应度值大于山峰适应度值，则将替换山峰值和点。循环上述步骤，直到找到邻域范围内的最大适应度值个体[18]。基于爬山算法基础，罗彪等提出了一种定向爬山算法[19]。牟宏鑫等针对图像调焦问题，提出了一种改进的自动调焦爬山算法，有效地避免了局部极值的干扰[20]。

(2) 模拟退火算法。模拟退火算法源于固体退火原理，每个解个体是一个分子，解的适应度值则相当于分子的内能。首先固体加热到非常高的温度，此时固体内部粒子内能增大，然后逐渐降低温度，粒子内能逐渐减小，当温度达到基态时内能最小，即是算法求到的最小解[21]。所以该算法的重要参数有初始温度、可降到的最低温度、控制温度降低的衰减因子和同一温度时的迭代次数等。

(3) 禁忌搜索算法。前面两种局部搜索算法，只比较了子代个体与父代个体间的最优解，这使算法容易陷入局部最优，而禁忌搜索算法为了避免局部最优，构建了禁忌搜索表，该表记录了出现过的最优的 n 个解，在新的搜索中，若新解优于表中的 n 个解，则替换。

3.2.3 Memetic 算法研究分类

近些年，Memetic 算法受到越来越多研究人员的关注，并被广泛应用于许多不同的领域，如最优化搜索、自动规划、机器人学习、经济模型、免疫系统、社会系统以及进化和学习的交互等。

从优化的观点来讲，针对某些问题，Memetic 算法比传统的遗传算法更有效率，它可以更快地找到最优解，同时求解的精度更高。因此 Memetic 算法获得了广泛的认同，特别是在组合优化问题上，采用 Memetic 算法的求解结果往往比其他的启发式搜索方法要好很多[34]。

Memetic 算法根据发展历程可以分为规范 Memetic 算法、适应性 Memetic 算法(超启发式 Memetic 算法、多 Memetic 算法、协同 Memetic 算法、后拉马克 Memetic 算法)等类型。

(1) 规范 Memetic 算法：遗传算法作为一种计算模型，充分模拟了生物的进化机制；而 Memetic 算法则是一种模拟文化进化的机制[35]，可以看作人们交流思想时被复制的信息单元。Memetic 算法中的种群是由许多的局部最优值组成的。

(2) 适应性 Memetic 算法：作为一个新的研究领域，Ong 等对其进行了很好的归纳，并称其为适应性 Memetic 算法[36]。适应性 Memetic 算法包括 Krasnogor 提出的多 Memetic 算法[37-40]，Smith 提出的与多 Memetic 算法相似的协同 Memetic 算法[41-42]，Ong 和 Keane 等提出的后拉马克 Memetic 算法以及超启发式 Memetic 算法等。这些算法具有一个共同的特点，那就是在进行搜索的过程中会采用很多密母，而具体采用哪个密母(如 meme、文化

基因)则是动态选择的。首先算法会随机或按照某种特定的方式生成种群,接下来针对种群中的每一个个体,在密母池中选择合适的密母进行局部搜索。针对组合优化问题的求解,Cowing 等提出了超启发式[43],它描述了融合不同的密母以在不同的确定点应用不同密母的思想。Krasnogor 等也提出了与超启发式类似的思想。针对超启发式搜索的局部搜索的方式也很多,拉马克和班德文学习正是密母算法中经常采用的学习机制。通过个体适应度值的局部改进,原来种群将被改进种群取代。Ong 和 Keane 在对连续非线性函数优化的过程中,采用了后拉马克学习机制对标准问题的特性进行了研究[43]。由于在对 Memetic 算法搜索的研究中主要采用了拉马克学习机制,因此这种 Memetic 算法被称为后拉马克学习。这种机制的机理主要是在多密母信息中引入竞争和合作,以更有效地解决问题。

3.3 基于 Memetic 算法的社区检测

复杂网络是复杂系统(比如万维网、在线社区、组织系统、商业系统、电网系统、神经系统、新陈代谢系统、食物链网络、合作系统、疾病控制系统、资源分配系统、交通系统、信息预测与推荐系统等)的重要研究方向[22-23],如何使用高效计算模型分析、挖掘和解译庞大数据中的潜在有用信息是复杂网络的重要研究内容。复杂网络不仅能有效展现网络单元(节点)之间相互作用的拓扑结构特性及其动态演化过程,还能够很好地表达复杂系统数据间的潜在关联及其功能特性。在复杂网络的表示方法中,节点(顶点)代表复杂系统的基本实体,边(链接)则表示系统中实体间的相互作用方式或关系[24-25]。

在生物系统的研究中,生物分子间复杂的相互作用被抽象为网络模型。分子网络描述了各种不同的生物活动进程是如何运作的,包括细胞间信号传递、新陈代谢和基因调控等,通常将这些生物分子网络看作一个系统来分析它们所完成的生物功能[26]。在社交网络中,把人作为节点,人与人之间的交流建模为边,通过对该网络的分析,可以把社交网络分成多个子网络(相当于社团)。同时还可以找到网络中最有影响力的节点,即找到社交网络中的核心人。

社区检测就是把复杂网络划分,找到具有共同点的网络节点,进一步研究网络性质。在实际应用中,社区检测可以为相同社区的用户提供更准确的服务,对电路系统、物流系统等进行全局优化,在资源分配和疾病控制上也能起到重要作用。用于社区检测的方法有很多,如传统的基于优化规则的贪心算法、多阶段贪心算法[78]、遗传算法等,这些算法的缺点均是容易陷入局部最优。下面将介绍 Memetic 算法在社区检测上的具体应用。Memetic 算法通过将全局搜索与局部搜索机制相结合的思路,在一定程度上缓解了陷入局部最优的问题。

3.3.1 多目标 Memetic 优化算法

在解决实际问题时,其目标往往不是单一的,而是多个目标的优化,如购买商品时则是希望它既好又便宜,所以多目标问题的求解非常重要,同时多目标优化不同于单目标优化时只存在一个解,多目标优化的解是一组互相无法比较优劣的解集。多目标 Memetic 算法的思想就是把多目标遗传算法全局优化的框架和一个或多个的局部优化算法混合,形成新算法。常用的求解多目标问题的遗传算法方法包括基于支配的 NSGA-Ⅱ[27] 和基于分解的 MOEA/D[28]。

1. 多目标社区检测的目标函数

对于一个无向网络，可以将它表示为 $G=(V,E)$，其中 V 表示节点集，E 表示网络中边的集合。网络的邻接矩阵为 $\boldsymbol{A}$，若节点 i 与节点 j 连接，则 $A_{ij}=1$，反之为 0。设有两个不相交的节点子集 V_1 和 V_2，即 $V_1 \bigcap V_2=\varnothing$、$V_1 \in V$、$V_2 \in V$。则模块密度 D 定义为：

$$D=\sum_{i=1}^{m}\frac{L(V_i,V_i)-L(V_i,\overline{V_i})}{|V_i|} \tag{3-1}$$

其中，$L(V_1,V_2)=\sum_{i\in V_1,j\in V_2}A_{ij}$，表示 V_i 节点集内部的所有边；$\overline{V_1}=V-V_1$，$L(V_i,\overline{V_i})$表示 V_i 内部节点到外部节点的边；$|V_i|$表示 V_i 内的节点数。将模块密度 D 差分成两部分，即可构成多目标 Memetic 优化算法的两个目标函数，第一个目标函数 NAR(Negative Ratio Association)反映了社区内节点连接的紧密程度，第二个目标函数 RC(Ratio Cut)反映了社区间的分离程度。将目标函数规范为最小化问题，则最终的多目标函数为：

$$\begin{cases}\min f_1=\mathrm{NAR}=-\sum_{i=1}^{m}\frac{L(V_i,V_i)}{|V_i|}\\ \min f_2=\mathrm{RC}=\sum_{i=1}^{m}\frac{L(V_i,\overline{V_i})}{|V_i|}\end{cases} \tag{3-2}$$

2. 解的表达

在解决社区检测类问题时，因为每个节点都有自己的社区编号，所以可以将个体 $\boldsymbol{x}_a$ 编码为：

$$\boldsymbol{x}_a=\{x_{a1},x_{a2},\cdots,x_{aN}\} \tag{3-3}$$

其中，x_{ai} 代表个体 $\boldsymbol{x}_a$ 对节点 v_i 赋予的标签，它的取值范围即为社区编号范围。若 $x_{ai}=x_{aj}$ 则认为节点 v_i 和 v_j 在同一编号的社区中，否则不在同一社区。

3.3.2 局部搜索

常用的局部搜索方法有模拟退火算法、爬山算法、禁忌搜索算法等。这里将介绍模拟退火算法和一种基于节点层、社区层、集群层的多层学习策略。

1. 模拟退火局部搜索算法

模拟退火算法不同于其他的局部搜索算法，它会以一定的概率保留较差的解，给解集带来多样性。其算法描述如表 3.1 所示。

表 3.1 模拟退火算法描述

PROCEDURE：模拟退火算法
Input：初始温度 T_e，退火因子 θ，解集 P，P_{best} While 不满足停止条件 $P'\leftarrow$ get neighbor(P)； If $F(P')>F(P_{\mathrm{best}})$ $P_{\mathrm{best}}\leftarrow P'$； Else if random(0,1)<

续表

$P_{best} \leftarrow P'$；
End if
$T_e \leftarrow T_e * \theta$；
End while
Output：新解集 P

其中，T_e 是初始温度，需要提前设置为一个较高的值；θ 是退火因子，用来控制该算法的退火速度，由表 3.1 可以看出，该值越大退火速度越慢；P_{best} 表示输入解集中适应度值最高的解(染色体)；P'表示输入解集的邻居解，在社区检测中，相当于新的社区划分，是与已有划分相似的邻居解；F 表示适应度函数，在基于分解的多目标遗传算法中，F 是子问题上的适应度函数。

2. 多层学习策略

多层学习策略是一种基于多层学习来改善个体解 $\boldsymbol{x}$ 的局部搜索方法。基于多层学习的局部搜索策略是根据网络节点、社区和集群结构的特定邻域知识而提出的，它由 3 个从低层次到高层次的学习策略组成。每一层次的学习策略都能够快速收敛到局部最优解。高层次学习策略能有效地帮助低层次学习策略跳出其所得到的局部最优解。

1) 节点层学习策略

节点层学习策略作用在网络中的每一个节点上，其过程如表 3.2 所示，其中 $\boldsymbol{x}_a$ 表示一个染色体。给定一个 N 节点的网络 G，节点层学习策略如下：首先生成一个随机序列 $R(\{r_1, r_2, \cdots, r_N\})$，然后根据序列中的顺序选择节点 v_{r_i}，利用标签传播规则或其他融合策略更新该节点的标签 x_{ar_i}。重复上述过程直到每个节点的社区标签都不改变为止。节点层学习策略有助于遗传算法快速收敛到搜索空间的局部最优解，同时遗传算法也可以帮助节点层学习跳出局部最优，去搜索下一个更优的局部最优解。

表 3.2 节点层学习算法流程

Procedure：节点层学习
Input：$\boldsymbol{x}_a$，网络 G 1. 产生一组随机序列 $R \leftarrow \{r_1, r_2, \cdots, r_N\}$； 2. 按照序列上面产生的序列 R，更新网络中节点 v_{r_i} 的社区标签； 3. 如果 $\boldsymbol{x}_a$ 的结果发生改变则转向步骤 1，否则停止运行； Output：$\boldsymbol{x}_a$

2) 社区层学习策略

社区层学习策略作用在前一层的解 $\boldsymbol{x}_a$ 上。假设染色体 $\boldsymbol{x}_a$ 划分出了 N_{x_a} 个社区，命名为社区 $C_1, C_2, \cdots C_{N_{x_a}}$。把每个社区里的所有点看成是一个合在一起的新点，即社区 C_1 是个新节点 v_1'，由此构造的点称为超级节点，并由超级节点构成新网络 G'。表 3.3 为社区层学习策略算法流程，其中函数表示进行节点层学习策略。需要注意的是超级节点间的连接问题，若 $v_i \in C_a$，$v_j \in C_b$ 且 v_i，v_j 间有连接，则认为由 C_a、C_b 形成的超级节点 v_a'、v_b'间有连接，且超级节点 C_a、C_b 间连接的权重为边数之和。

表 3.3 社区层学习算法流程

Procedure：社区层学习
Input：$\boldsymbol{x}_a$，网络 G 1. 构造超级节点 $V'=\{v'_1,v'_2,\cdots,v'_{N_{x_a}}\}$，构造新网络，生成新的邻接矩阵 $\boldsymbol{A}'$； 2. 给超级节点赋社区标签，即 $x_{s_i}=i$，$1\leqslant i\leqslant N_{x_a}$，得到新网络的初始化解 $\boldsymbol{x}_s$； 3. $\boldsymbol{x}_s\leftarrow$NodelLearning($\boldsymbol{x}_s$，$G'$)； 4. 解码 $\boldsymbol{x}_s$ 得到原网络 G 的新解 $\boldsymbol{x}_e$； Output：$\boldsymbol{x}_e$

社区层学习策略也用来帮助节点层学习策略跳出局部最优解，但是因为错误被合并在一个社区的节点，无法在该层学习中被分开，所以这两层的学习策略依然容易陷入局部最优解。

3）集群层学习策略

集群层学习策略的过程如表 3.4 所示。它作用在两个染色体 $\boldsymbol{x}_g$、$\boldsymbol{x}_e$ 上，其中 $\boldsymbol{x}_g=\{x_{g1},x_{g2},\cdots,x_{gN_{x_g}}\}$是种群中的最优染色体，$\boldsymbol{x}_e=\{x_{e1},x_{e2},\cdots,x_{eN_{x_e}}\}$是前一层社区层学习产生的解。集群层学习策略包含两个阶段。第一阶段为，对 $\boldsymbol{x}_g$ 中的每一个划分的社区 C_i，$1\leqslant i\leqslant N_{x_g}$ 所对应的节点，根据下面两个原则重新分配到社区中。

（1）如果节点 v_j、v_k 在解 $\boldsymbol{x}_g$、$\boldsymbol{x}_e$ 上被划分为同一社区内，则在新解上被分到该社区中；

（2）如果节点 v_j、v_k 在 $\boldsymbol{x}_g$ 中被划分在同一社区，而在 $\boldsymbol{x}_e$ 中被划分为不同社区，则在新解中也被划分到不同社区内。

定义第一阶段返回的解为 $\boldsymbol{x}_f=\{x_{f1},x_{f2},\cdots,x_{fN_{x_f}}\}$。第二个阶段是在 $\boldsymbol{x}_f$ 上使用前两层学习策略以便得到更好的社区划分。在表 3.4 中，函数 NodeLearning（）和 CommunityLearning()分别表示节点层学习策略和社区层学习策略。最终集群层学习策略将返回一个新的染色体 $\boldsymbol{x}_{l'}$。

表 3.4 集群层学习策略

Procedure：集群层学习
Input：$\boldsymbol{x}_g$，$\boldsymbol{x}_e$，网络 G 1. 设置 $x_{fi}\leftarrow 0$，$i=1,2,\cdots,N$，$c\leftarrow 0$，$i\leftarrow 1$； 2. 确定社区 C_i 的节点 $C_i=\{v_{i_1},v_{i_2},\cdots,v_{i_{\lvert C_i\rvert}}\}$，其中 $\lvert C_i\rvert$ 是社区 C_i 内的节点规模； 3. 对 C_i 中的每个节点 v_{i_j}，如果 $x_{fi_j}=0$，则 $c\leftarrow c+1$，$x_{f_k}\leftarrow c$，其中 k 满足 $\forall k\in\{v_k\in C_i \& x_{ei_j}=x_{ek}\}$。否则转向步骤 4； 4. 如果 $i\leqslant N_{x_g}$，则 $i\leftarrow i+1$ 并转向步骤 2，其中 N_{x_g} 是 $\boldsymbol{x}_g$ 中的社区数。否则转向步骤 5； 5. $\boldsymbol{x}_f\leftarrow$NodeLearning($\boldsymbol{x}_f$，$G$)； 6. $\boldsymbol{x}_{l'}\leftarrow$communitylearning($\boldsymbol{x}_f$，$G$)； Output：$\boldsymbol{x}_{l'}$

集群层学习策略能够帮助前两层学习策略跳出局部最优解。首先，前两层学习策略陷入局部最优的主要原因在于被合并的多个节点和社区很难被再次分开[29]。而集群层学习策略就可以解决该问题，分开合并的节点和社区。同时重组的社区还可以获得更高的模块

度。其次，集群层学习策略是作用于两个解上的集成聚类技术。新解将得到这两个解上相同的部分，不同的部分被分开。因此，集群层学习策略收集了来自两个或更多解的社区模式结构特性。

由上述两种局部搜索机制可以看出，局部搜索有助于在全局搜索过程中找到当前最优解的邻域内优于该个体的新解，加快了算法的收敛速度，同时优良的局部搜索机制还可以改善算法本身容易陷入局部最优解的情况，如上述提到的多层学习策略，多层机制增大了邻域搜索的能力，增加了新解的多样性，因此更有可能找到最优解。

3.4 基于 Memetic 算法的限量弧路由问题

在对 CARP 求最优解的过程中，随着问题规模的增加，解空间的规模会成指数增长，所以又有了大规模的限量弧路由问题（Large-Scale Capacitated Arc Routing Problem，LSCARP）。在车辆路径规划这一问题中，通常任务数目都比较多。与其他算法相比，元启发式算法能够在给定的时间内找到问题的次优解，但是计算需求量很大，花费的时间更多。从算法的角度出发，Memtic 算法可以使算法跳出局部最优，获得一个更好的解。

为了解决 LSCARP，2014 年提出的 RDG-MAENS 采用了一种分治策略[30-31]，把问题分解成很多子问题，分别予以解决，然后再由子问题的解得到完整的解。2015 年，有研究者针对 RDG-MAENS 中个体所属子种群的分配方式不合理问题，提出了一种改进的算法 IRDG-MAENS，收敛性有很大提高。[32]

3.4.1 路由距离分组

RGD-MAENS 方法采用路由距离分组（Route Distance Grouping，RGD）分解方法将 CARP 问题的任务集进行分组，使其分解成多个小的子任务集进行求解。这个分组就是路由距离分组，首先定义两个节点之间的距离，这里使用的不是地理信息中的欧几里得距离而是顶点间的路由距离，再利用迪杰斯特拉算法确定两个节点间的最短距离，最后根据路由中所有任务节点的距离确定路由间的距离。两个路由间的距离越小，说明这两个路由越相似，越应该放到一组中。基于这样的定义，分组就变成了对路由间的聚类，两个路由间的距离越小就更可能被归为同一类，这里聚类运用了模糊 K-mediods[33] 方法。聚类的关键是路由组中心点的确定，确定好最优的路由中心，才能将相似的路由放到一起进化，节省计算资源，加快收敛速度，提高解的质量。先定义两个任务 z_1 和 z_2 的亲密度为：

$$(z_1, z_2) = \frac{\sum_{i=1}^{2}\sum_{j=1}^{2}\Delta[v_i(z_1), v_j(z_2)]}{4} \tag{3-4}$$

其中，$\Delta[V_i(z_1), V_j(z_2)]$ 是任务 z_1 第 i 个终点和任务 z_2 第 j 个终点之间的距离。因此 $\Delta_{\text{task}}(z_1, z_2)$ 就是 z_1 和 z_2 间可能距离的平均值。基于任务距离，s_1 和 s_2 之间的距离定义为：

$$\Delta_{\text{route}}(s_1, s_2) = \frac{\sum_{z_1 \in s_1}\sum_{z_2 \in s_2}\Delta_{\text{task}}(z_1, z_2)}{|s_1| \cdot |s_2|} \tag{3-5}$$

因此 $\Delta_{\text{route}}(s_1,s_2)$ 是任务 z_1 和 z_2 间所有的任务对 s_1 和 $s_2(s_1\in z_1,s_2\in z_2)$ 间的平均距离。再把 $\Delta_{\text{route}}(s_1,s_2)$ 根据 $\Delta_{\text{route}}(s_1,s_1)$ 和 $\Delta_{\text{route}}(s_2,s_2)$ 进行规范化，最终得到任意路由 s_1 和 s_2 间的距离为：

$$\hat{\Delta}_{\text{route}}(s_1,s_2)=\frac{\Delta_{\text{route}}(s_1,s_2)}{\Delta_{\text{route}}(s_1,s_1)}\cdot\frac{\Delta_{\text{route}}(s_1,s_2)}{\Delta_{\text{route}}(s_2,s_2)} \tag{3-6}$$

3.4.2 子问题解的更替

对于 LSCARP，其需要分解的数量很大，并且分解后得到的子问题仍然是 NP-hard 问题。针对这一情况，梅一提出了 RDG 分解方案。但是，每一代的子问题分解得到的解并不是最好的解，因为 RGD 没有根据当前群体信息动态地为各个子问题重新分配解，而只是将搜索到的最好的解进行保留，并不参与其他子问题的求解。

在 IRDG-MAENS 算法中，首先根据 RDG 算法将大规模的问题进行分解，再对分解后得到的子问题单独求其最优解。在获得子问题最优解后立即用较好的解进行更替并通过邻域共享参与当前子代问题的求解。这不仅能够加快算法的收敛速度，还及时更替每个子问题较好的解，参与该循环和其他子问题的求解，这种方法更利于邻域共享和获得潜在的更好的解。表 3.5 是更加详细的子问题解的及时更替过程。

表 3.5 子问题解的及时更替

Procedure：子问题解的及时更替

1. for $i=1\rightarrow g$
   2. 采用 RDG 分解方案将任务集 Z 分解为 g 个子任务集合$(Z_1,Z_2,\cdots,Z_g)$，并对每个子任务集合单独求解；
   3. 产生也给子种群 $SP(z_i)$，采用 MAENS 算法求解该子种群，获得一个可行解 $s(z_i)$；
   4. 对所有的可行解 $s(z_i)$ 进行适应度排序，找到最好的可行解；
   5. 将获得的子问题中最好的可行解 $\bar{s}(z_i)$ 组成一个新的子种群，进化该子种群获得新的可行解 $s^*(z_i)$；
   6. 如果新的可行解 $s^*(z_i)$ 优于 $\bar{s}(z_i)$，则更新可行解；
   7. 如果 $i<g$ 则跳转执行步骤 2，采用最新的路由信息去分解任务集合 Z；
8. end for

3.4.3 基于分解的 Memetic 算法

遗传算法部分可以采用多目标进化算法(MOEA/D)的算法框架，其思想是，将一个多目标问题通过一定的方法分解为多个子问题，然后同时对这些子问题进行优化，在优化过程中，将考虑其邻域信息，实现在对子问题的优化过程中找到原多目标问题的非支配解。

假设多目标优化函数为 $G(x)=[G_1(x),G_2(x)]$，通过二维权重系数 $\alpha_i=(\alpha_{i1},\alpha_{i2})$ 将原多目标问题划分为多个单标优化问题。其中 α_{i1} 是目标函数 $G_1(x)$ 的权重系数；α_{i2} 是目标函数的权重系数。于是，分解后的单目标优化问题的函数可表示为：

$$g_i(x)=\alpha_{i1}G_1(x)+\alpha_{i2}G_2(x) \tag{3-7}$$

假设有 n 个由两个权重系数构成的权重向量，那么就可以将原多目标问题分解为 n 个子问题。每个子问题都有一个初始解。计算权重间的欧几里得距离，根据该距离为每个子

问题选择 T 个邻居，第 i 个子问题的邻居记为 $B(i)=\{i_1,i_2,\cdots,i_T\}$。从 $B(i)$ 中随机选择两个个体进行遗传操作产生新解 y，再更新邻域解：对于所有 $j\in B(j)$，如果 $g_i(x^j)<g_i(y)$，则 $x^j=y$。

参考文献

[1] Ong Y S, Lim M H, Tan K C. A multi-facet furvey on memetic computation[J]. IEEE Transactions on Evolutionary Computation, 2011, 15(5): 591-607.

[2] Ong Y S, Meng H L, Chen X. Research frontier: memetic computation-past, present & future [J]. IEEE Computational Intelligence Magazine, 2010, 5(2): 24-31.

[3] Moscato P. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms [J]. Caltech Concurrent Computation Program, C3P Report, 1989, 826: 1989.

[4] Gong M, Fu B, Jiao L, et al. Memetic algorithm for community detection in networks. [J]. Physical Review E, 2011, 84(5): 056101.

[5] Ma L, Gong M, Liu J, et al. Multi-level learning based memetic algorithm for community detection [J]. Applied Soft Computing, 2014, 19(2): 121-133.

[6] Gong M, Fu B, Jiao L, et al. Memetic algorithm for community detection in networks[J]. Physical Review E, 2011, 84(5): 056101.

[7] Ma L, Gong M, Liu J, et al. Multi-level learning based memetic algorithm for community detection [J]. Applied Soft Computing, 2014, 19: 121-133.

[8] Sun Y, Du H, Gong M, et al. Fast computing global structural balance in signed networks based on memetic algorithm[J]. Physica A: Statistical Mechanics and its Applications, 2014, 415: 261-272.

[9] Lacomme P, Prins C, Ramdane-Chérif W. A genetic algorithm for the capacitated arc routing problem and its extensions[C]//Workshops on Applications of Evolutionary Computation. Springer Berlin Heidelberg, 2001: 473-483.

[10] Lacomme P, Prins C, Ramdane-Cherif W. Competitive memetic algorithms for arc routing problems [J]. Annals of Operations Research, 2004, 131(1-4): 159-185.

[11] Shang R H, Jiao L C, Liu F. A novel immune clonal algorithm for mo problems [J]. IEEETransactions on Evolutionary Computation, 2012, 16(1): 35-49.

[12] Mei Y, Li X D, Yao X. Cooperative co-evolution with route distance grouping for large-scale capacitated arc routing problems[J]. IEEE Transactions on Evolutionary Computation, 2014, 18(3): 435-449.

[13] Mei Y, Tang K, Yao X. Decomposition-based memetic algorithm for multi-objective capacitated arc routing problems[J]. IEEE Transactions on Evolutionary Computation, 2011, 15 (2): 151-165.

[14] O'Hara T, Bull L. A memetic accuracy-based neural learning classifier system[C]//IEEE Congress on Evolutionary Computation, 2005: 2040-2045.

[15] Abbass H A, et al. A Memetic Pareto Evolutionary Approach to Artificial Neural Networks. The Australian Joint Conference on Artificial Intelligence, Adelaide, 2001.

[16] Bonfim T R, Yamakami A. Neural Network Applied to the Coevolution of the Memetic Algorithm for Solving the Makespan Minimization Problem in Parallel Machine Scheduling [J]. IEEE Transactions on Magnetics, 2008, 44(6): 782-785.

[17] Gong M, Fu B, Jiao L, et al. A memetic algorithm for community detection in networks[J]. Physical Review E, 2011, 84(5): 056101.

[18] 何军良,宓为建,严伟. 基于爬山算法的集装箱堆场场桥调度[J]. 上海海事大学学报,2007,28(4):11-15.

[19] 罗彪,郑金华,杨平. 基于定向爬山的遗传算法[J]. 计算机工程与应用,2008,44(6):92-95.

[20] 牟宏鑫,吴庆畅,张翠,等. 一种改进的自动调焦爬山搜索算法[J]. 昆明冶金高等专科学校学报,2010,26(03):32-35.

[21] 谢云. 模拟退火算法综述[J]. 微计算机信息,1998(5):7-9.

[22] Schneider C M,Moreira A A,et al. Mitigation of malicious attacks on networks[J]. Proceedings of the National academy of Sciences of the United States of America,2011,108(10):3838-3841.

[23] Watts D J,Strogatz S H. Collective dynamics of 'small-world' networks[J]. Nature,1998,393(6684):440-442.

[24] Strogatz S H. Exploring complex networks[J]. Nature,2001,410(6825):268-276.

[25] Newman M E. The structure and function of complex networks[J]. SIAM Review,2003,45(2):167-256.

[26] Koyutürk M. Algorithmic and analytical methods in network biology[J]. Wiley Interdisciplinary Reviews:Systems Biology and Medicine,2010,2(3):277-292.

[27] Deb K,Agrawal S,Pratap A,et al. A fast and elitist multi-objective genetic algorithm:NSGA-Ⅱ[J]. IEEE Transactions on Evolutionary Computation,2002,6(2):182-197.

[28] Zhang Q,Li H,MOEA/D:A Multiobjective Evolutionary Algorithm Based on Decomposition,IEEE Transactions on Evolutionary Computation,2007,11(6):712-731.

[29] Rosvall M,Axelsson D,BERGSTROM C T. The map equation[J]. The European Physical Journal Special Topics,2009,178(1):13-23.

[30] Omidvar M N,Li X,Mei Y,et al. Cooperative co-evolution with differential grouping for large scale optimization[J]. IEEE Transactions on Evolutionary Computation,2014,18(3):378-393.

[31] Li X,Yao X. Cooperatively coevolving particle swarms for large scale optimization[J]. IEEE Transactions on Evolutionary Computation,2012,16(2):210-224.

[32] Shang R,Dai K,Jiao L,et al. Improved memetic algorithm based on route distance grouping for multiobjective large scale capacitated arc routing problems[J]. IEEE Transactions on Cybernetics,2016,46(4):1000-1013.

[33] Krishnapuram R,Joshi A,Yi L. A fuzzy relative of the k-medoids algorithm with application to web document and snippet clustering[C]//IEEE International Fuzzy Systems Conference Proceedings,1999:1281-1286.

[34] Merz P,Freisleben B. A comparison of memetic algorithms,tabu search,and ant colonies for the quadratic assignment problem[C] //Proceeding. Int. Congr. Evol. Comput.,Washington DC,1999:2063-2070.

[35] Dawkins R. The Selfish Gene[M]. New York:Oxford Univ. Press,1976.

[36] Ong Y S,Lim M H,Zhu N,et al. Classification of Adaptive Memetic Algorithms:A Comparative Study[J]. IEEE Transactions On Systems,Man and Cybernetics-Part B,2006,36(1):141-152.

[37] Krasnogor N. Coevolution of genes and memes in memetic algorithms[C] //Proceeding. Genetic and Evol. Comput. Conf. Workshop Program,1999.

[38] Krasnogor N,Smith J. A memetic algorithm with self-adaptive local search:TSP as a case study[C] //Proceedings of theAnnual Conference on Genetic and Evolutionary Computation,San Francisco,CA,2000:987-994.

[39] Krasnogor N,Smith J. Emergence of profitable search strategies based on a simple inheritance mechanism[C] //Proceedings of the Annual Conference on Genetic and Evolutionary Computation,San Francisco,CA,2001:432-439.

[40] Krasnogor N. Studies in the theory and design space of memetic algorithms[D]. Bristol: University of West England,2002.

[41] Burke E, Smith A. Hybrid evolutionary techniques for the maintenance scheduling problem[J]. IEEE Trans. Power System,200,15(1): 122-128.

[42] Hansen P, Mladenovi'c N. An introduction to variable neighborhood search[M]//Glover F, Kochenberger G A. Handbook of Metaheuristics. Boston,MA: Springer,1998: 145-184.

[43] Ong Y S,Keane A J. Meta-Lamarckian Learning in Memetic Algorithm[J]. IEEE Transactions On Evolutionary Computation,2004,8(2): 99-1104.