

基于扩展邻域搜索问题的模因算法

IEEE成员柯唐、学生IEEE成员李美、IEEE研究员新耀

摘要: 有电电路路径问题 (CARP) 由于其在现实生活中的广泛应用, 在过去几年中受到了人们的广泛关注。由于CARP是np困难的方法, 精确方法只适用于小实例, 因此启发式和元启发式方法在求解CARP时被广泛采用。本文提出了一种模因算法, 即具有扩展邻域搜索 (MAENS) 的模因算法。MAENS不同于现有的方法, 在利用一种新的局部搜索算子, 即合并-分裂 (MS)。MS算符能够使用大的步长进行搜索, 因此有潜力更有效地搜索解空间, 并且不太可能陷入局部最优。实验结果表明, MAENS优于许多最先进的算法, 而MAENS的先进性能主要归功于MS算子。MS操作符的应用并不限于MAENS。它可以很容易地推广到其他方法中。

索引术语- 有能力的电电路由问题 (CARP), 进化优化, 局部搜索, 模因算法, 元启发式搜索。

I. 介绍

他的ARC路由问题是一个经典的问题, 在现实世界中有许多应用, 如城市废物收集, 后交付, 打磨或盐渍街道 [1], [2] 等。这是一个组合优化问题, 需要确定受 [3] 约束的车辆的最小成本路由计划。本文研究了电电路由问题的最典型形式是有容量电电路由问题 (CARP)。它可以描述为: 一个混合图 $G = (V, E, A)$, 有一组顶点用 V 表示, 一组用 E 表示的边和一组弧 ($i.e.$, 有向边), 用 A 表示。有一个中央仓库顶点 $v \in V$, 其中有一组车辆的基础。一个子集 $E' \subseteq E$ 由所有需要的边组成

稿件于2008年11月18日收到; 2009年2月14日修订; 2009年5月5日接受。第一版出版于2009年8月11日; 当前版本发布于2009年9月30日。这项工作是由工程和物理科学研究委员会授予EP/E058884/1 “动态优化问题的进化算法: 设计、分析和应用,” 中国国家自然科学基金会资助60533020和U0835002, 由外国学者在大学研究和教学项目, 授予B07033。

K. 唐和Y. 梅就职于中国科技大学计算机科技学院自然灵感计算与应用实验室, 中国, 合肥230027 (电子邮件: 基张@ustc.edu.cn; meiyi@mail.ustc.edu.cn)。

X. 姚在伯明翰大学计算机科学学院计算智能与应用研究卓越中心工作。K. (电子邮件: x.yao@cs.bham.ac.uk)。

数字对象标识符10.1109/TEVC.2009.2023449

服务和子集 $ar \subseteq A$ 也给出了所有需要提供的弧组成。这两个子集的元素分别被称为边缘任务和圆弧任务。图中的每条边或弧都与需求、服务成本和报废成本 (车辆沿边/弧行驶而不提供服务的成本) 相关联。对于不需要服务的边缘和弧线, 需求和服务成本都为零。解决该问题的一个方案是由多个车辆路线组成的路由计划, 其目标是在以下约束下将路由计划的总成本最小化:

- 1) 每条路线都在车站开始和结束;
- 2) 每个任务都只在一条路径上提供服务;
- 3) 各路线的总需求不得超过车辆的容量 Q 。

由于CARP是np-硬 [4], 精确的方法只适用于小型实例。因此, 在文献中经常考虑启发式和元启发式。例如, 增强-合并 [4]、路径扫描 [5]、[6] 中提出的“路由第一集群第二”型启发式, 以及Ulusoy的启发式 [7] 是典型的启发式方法。2000年, 赫兹等人。建议对鲤鱼 (地毯) [8] 进行标签搜索。在地毯中, 一个解决方案被表示为一组路由, 每个路由都是一个有序的顶点列表。每个顶点都与一个01变量相关联, 指示该顶点和连续顶点之间的边。在地毯的基础上, 提出了一种可变邻域下降 (VND) 算法, 将地毯中的标签搜索过程替换为可变邻域搜索过程 [9]。2003年, 博伦斯等人。开发了一种针对鲤鱼 [10] 的引导局部搜索 (GLS) 算法。GLS采用了一种边缘标记方案, 它根据之前的搜索过程的信息来标记 (未标记) 边缘。局部搜索操作符只应用于那些已标记的边缘, 以便使搜索过程更有效。之后, 拉科姆等人。提出了一种模因算法¹⁾, 它结合了遗传算法 (GA) 和局部搜索 [11]。LMA采用了一种基因型编码方案。也就是说, 将一个解决方案表示为一系列任务, 并省略两个后续任务之间的中间顶点。为了从一个解决方案中获得完整的路由计划, 需要将每两个后续任务与它们之间的最短路径连接起来, 然后应用Ulusoy启发式将序列分离为多个路由。

¹在文献中, MA通常被称为一个通用的框架, 而不是任何特定的算法。为了清晰起见, 我们表示由Lacomme等人提出的MA。[10] 为LMA。

2006年, Handa等人. 提出了一种进化算法 (EA) [1], [2] 的盐析路线优化问题. K. 该EA采用了与LMA, i相似的编码方案. e., 一个解决方案由一系列任务表示. 不同之处在于, 由EA得到的解可以自然地分为不同的路径, 因此没有必要再使用Ulusoy的启发式了. 此外, EA和LMA还采用了不同的进化算子和评价方案. 最近, 布兰度和Eglese提出了一种针对鲤鱼的确定性禁忌搜索算法 (TSA) [12]. 在他们的工作中, 描述了TSA的两种变体, 即TSA1和TSA2. TSA1是一种相当标准的标签搜索算法, TSA2可以看作是应用TSA1与许多不同的初始解. 实验研究表明, TSA2的TSA2优于TSA1, 且优于地毯和LMA. 在[13]中, 我们提出了一个全球修复操作符 (GRO), 它旨在修正低成本的不可行的解决方案. 结果表明, GRO与TSA1结合可以显著提高解质量, 甚至可以加速算法的收敛速度. 到目前为止, 我们已经引入了启发式和元启发式两种方法. 可以看出, 前者在早期很流行, 而后者最近吸引了越来越多的兴趣. 此外, 许多现有的元启发式方法在他们的框架中包含了以前的启发式方法. 例如, crate、LMA和TSA都使用一个或多个上述启发式方法来获得初始解, 然后尝试进一步改进它们. 因此, 元启发式方法在解决方案质量方面通常优于启发式方法也就不足为奇了, 尽管它们在计算上更昂贵. 幸运的是, 功能强大的现代计算机可以很容易地支付额外的计算成本. 在本文中, 我们在MA的框架内研究了鲤鱼. 作为进化计算的一个新兴领域, MAs是一种基于种群的元启发式搜索方法, 它结合了全局搜索策略 (e. g., 在许多不同的名称下, 已经被研究过, 如鲍德文EAs, 拉马克EAs, 文化算法, 遗传局部搜索等. 据报道, 它们不仅收敛于高质量的解决方案, 而且比传统的EAs更有效地搜索. MAs的成功已经被揭示在各种现实世界的问题[14]–[16]上, 包括CARP, 正如LMA [11]所证明的那样. 与传统的EAs相比, MAs的成功存在两个关键问题. 一种是全局搜索和局部搜索之间的适当平衡, 另一种是本地搜索的成本效益协调. 因此, 局部搜索过程, 通常是为了利用感兴趣的问题的领域知识, 在MAs中起着最重要的作用. 在CARP的环境中, 局部搜索通常是通过一些传统的移动操作符来进行的, 如单插入、双插入、交换等. [11]. 这些移动操作符只修改了当前解决方案的一小部分. 更直观地说, 它们可以说有很小的搜索步长, 并在当前解决方案的一个小范围内进行搜索. 有了这些特性, 这些算子可以期望在具有少量局部最优和解空间小的简单问题上表现良好. 然而, 他们

当解空间变得很大或包含许多局部最优时, 或者当解空间由分离的可行区域组成时, 可能不再工作. 在这种情况下, 大步长的局部搜索可能更可取, 或者跳出局部最优, 或者从一个可行区域跳到另一个可行区域, 或者更有效地进行搜索. 从一般的优化角度来看, [17]从理论上讨论了大步长的好处, 证明了较大邻域的模拟退火 (SA) 优于较小邻域的SA. 不幸的是, 步长问题本身很少在CARP的背景下得到解决, 更不用说设计一个改进的模因方法了. 基于上述考虑, 本文提出了一种新的CARP移动算子, 合并-分裂 (MS) 算子. 与传统的移动算子相比, MS算子具有较大的搜索步长, 这是可变的. 因此, 它可以灵活地在在一个解决方案的大邻域内进行局部搜索. 我们将MS算子纳入MA框架, 并开发了具有扩展邻域搜索 (MAENS) 的模因算法

挑剔MAENS已经在四组鲤鱼上进行了评估基准测试实例 (总共181个实例), 并与现有的5种元启发式算法进行了比较. e., 地毯[8], VND [9], GLS [10], LMA [11], 和TSA2 [12]. 实验结果表明, MAENS在局部最优的困难实例上优于现有五种算法, 在简单实例上, 都达到了全局最优.

本文的其余部分组织如下. 第二节介绍了本文的初步背景, 包括CARP的形式化问题定义、MA的一般框架, 以及局部搜索的传统移动算子. 第三节详细描述了MS操作符. 在此之后, 在第四节中提出了MAENS. 第五节介绍了实验研究, 其中包括MAENS和其他算法之间的实证比较, 以及MS操作符的有效性的演示. 最后, 结论和未来的工作将在第六节中介绍.

II. 背景

本节介绍了本文的背景资料. 我们从鲤鱼的符号、解表示和数学表示开始, 然后简要描述了鲤鱼的一般框架和传统的移动算符.

A. 符号和问题的定义

CARP包括寻求车辆的最低成本路径计划, 以满足所有所需的边缘 $E \subseteq E$ 和必需的圆弧线 ar 给定图 $G = (V, E, A)$ 的 $\subseteq A$, 受到一些约束. 每个圆弧任务都被分配了一个唯一的ID, 比如 t . 每条边 (i, j) 被认为是一对弧, $\langle i, j \rangle$ 和 $\langle j, i \rangle$, 每个方向对应一个. 因此, 每个边缘任务被分配两个id. 为了方便起见, id被设置为正整数. 每个ID t 与五个特征相关联, 即尾部 (t)、头部 (t)、 $sc(t)$ 、 $dc(t)$ 和 $dem(t)$, 代表尾部和头部顶点、服务成本、终端成本, 以及

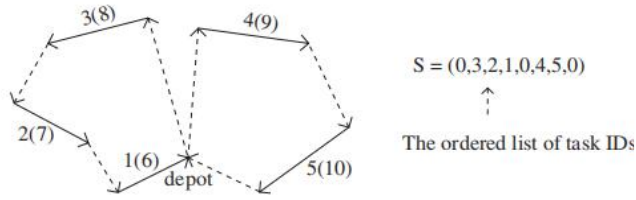


图1. 解决方案表示法的说明。括号中的id表示当前方向的倒置。

- | | |
|---|------------------------|
| 1 | 初始化：生成一个初始填充 |
| 2 | 当停止标准不满足时，请这样做 |
| 3 | 评估人群中的所有个体 |
| 4 | 使用对每个个体的进化操作符来进化一个新的种群 |
| 5 | 以概率进行局部搜索 PLS ; |
| 6 | 最后部分 |
| 7 | |
| 8 | 端 |

图2. MAs的一般框架。

分别是对对应任务的需求。如果 t 属于一个边缘任务，设 $inv(t)$ 表示任务 t 的反演。服务成本、任务成本(t)需求分别与 $sc(t)$ 、 $dc(t)$ 和 $dem(t)$ 相同。但是请注意，每个边缘任务只应该在任何一个方向(i. e., 只提供一个任务 t 和 $inv(t)$)。为了在解决方案中的不同路由，我们还定义了虚拟任务。虚拟任务的尾部顶点和头部顶点都是仓库顶点深度，其ID设置为0。像许多其他现有的元启发式方法一样，我们将CARP的解决方案表示为一个有序的任务列表(id)，用 $S = (S_1, S_2, \dots, K)$ ，其中 i 是 S 的第 i 个元素(任务)。图1给出了这种解决方案表示的一个简单说明。给定解 S ，通过将每两个后续任务之间的最短路径连接起来，可以得到相应的路由计划。e., 找到从前一个任务的尾顶点到后续任务的头顶点的最短路径)，这可以很容易地通过Dijkstra的算法[18]找到。在CARP中，术语“最短路径”等价于具有最小无死代价的路径。Let $sp(S_i, S_{i+1})$ 表示之间最短路径的总去死代价 i 和 $i+1$ ，则 S 的总成本可以写成

$$TC(S) = \sum_{i=1}^{length(S)-1} [sc(S_i) + sp(S_i, S_{i+1})] \quad (1)$$

其中，长度(S)表示序列 S 的长度。

从图1可以看出，每个解决方案 S 可能由多条路由组成(e. g., 图中的例子。1有两条路线)，每条路线都从车站开始和结束。因此，我们可以进一步用这个形式来写 S

$$S = (R_1, R_2, K, R_m) \\ = (0, \underbrace{R_{11}, R_{12}, K, 0}_{R_1}, \underbrace{R_{21}, R_{22}, K, 0}_{R_2}, \dots, \underbrace{R_{m1}, R_{m2}, K, 0}_{R_m}) \quad (2)$$

其中 m 是 S 和每个路线的路线数 R 表示一条路线。显然，每一个 R 也包括一个

任务的子序列，以及负载(i. e., 总需求)是

$$load(R_i) = \sum_{k=1}^{length(R_i)} dem(R_{ik}). \quad (3)$$

鉴于上述所有符号和上述鲤鱼的三个约束条件，我们现在得出以下对鲤鱼的表示：

$$\begin{aligned} \min_S TC(S) &= \sum_{i=1}^{length(S)-1} [sc(S_i) + sp(S_i, S_{i+1})] \\ s.t. : app(S_i) &= 1, \forall S_i \in A_R \\ app(S_i) + app(inv(S_i)) &= 1, \forall S_i \in E_R \\ m &\leq nveh \\ load(R_i) &\leq Q \end{aligned} \quad (4)$$

其中 $app(S_i)$ 计算该任务的时间 i 出现在整个序列中， $nveh$ 是在仓库中可用的车辆的数量，而 Q 是车辆的容量。

B. 模因算法的总体框架 (MAs)

1989年由Moscato于1989年[19]首次提出，MAs受到达尔文自然进化原理和道金斯的模因[20]概念的启发。从进化计算的角度来看，MAs可以被视为一种基于种群的EAs形式，与能够执行局部细化[19]的个体学习程序杂交。在不失一般性的情况下，MAs的框架可以概括为图2。

从图2可以看出，MAs和传统的EAs之间的一个主要区别是，EAs的突变操作符被MAs中的局部搜索所取代。因此，MAs的成功在很大程度上是归功于对本地搜索操作符的适当采用，而在MAs的增量开发中，许多重要的工作都围绕着本地搜索过程[21]–[24]也就不足为奇了。进化算符通常非常一般，适用于各种问题，局部搜索算符通常被期望包含一些领域特定领域的启发式，以便MAs可以很好地在通用性和问题特异性之间取得平衡。例如，局部启发式方法或传统的精确枚举方法，如单纯形法、牛顿/拟牛顿法、共轭梯度法和线搜索法，是典型的数值优化的局部搜索策略。在组合优化的背景下，局部搜索方法通常是专门设计来很好地服务于一个感兴趣的问题，e. g., k -基因交换，旅行推销问题的 k -opt算法，等等。在下一节中，我们将简要介绍一些传统的鲤鱼本地搜索操作符。

C. 传统的移动本地搜索操作符

回想一下，CARP的解决方案是编码为任务id序列的，因此通常通过应用移动操作符来对候选解决方案进行局部搜索。在

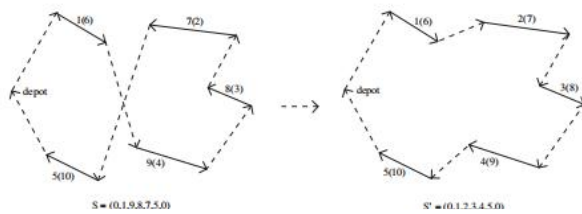


图3. 2-选择一条单一的路线。

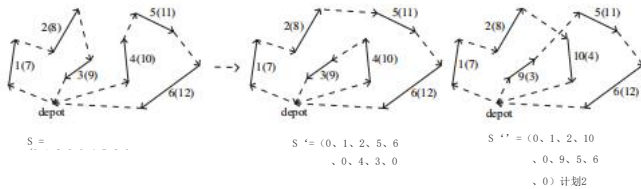


图4. 2-选择双路线。

文献中, CARP有四种常用的移动操作符, 即单插入、双插入、交换和2-选择[11]。

1) 单个插入: 在单个插入移动中, 任务从当前位置删除, 并重新插入到当前解决方案的另一个位置或新的空路由。如果所选任务属于一个边缘任务, 则在将任务插入到“目标位置”时, 将考虑它的两个方向。将选择导致更好的解决方案的方向。

2) 双插入: 双插入移动类似于单个插入, 除了移动了两个连续的任务, 而不是单个任务。与单个插入类似, 对于边缘任务都考虑了两个方向。

3) 交换: 在交换移动中, 选择两个候选任务并交换它们的位置。与单个插入类似, 对于边缘任务都考虑了两个方向。

4) 2-opt: 有两种类型的2-选择移动操作符, 一种是单路线, 另一种是双路线。在一个单一路线的2-opt移动中, 一个子程序(i.e., 选择了路线的一部分), 其方向被反转。当将2-opt移动应用到双路由时, 每条路由首先被切割成两个子线, 并通过重新连接这四个子线产生新的解决方案。图3和4分别说明了这两个2个可选择的移动操作符。在无花果。3、给定一个解 $S = (0, 1, 9, 8, 7, 5, 0)$, 选择从任务9到任务7的子集, 使其方向反转。在无花果。4、给定解决方案 $S = (0, 1, 2, 3, 0, 4, 5, 6, 0)$, 第一条路由在任务2和3之间切割, 第二条路由在任务4和5之间切割。通过将任务2与任务5连接, 任务4与任务3连接, 或者将任务2与任务4与任务5的反转连接, 可以得到一个新的解决方案。在实践中, 人们可以选择成本较小的一种。与前三个运算符不同, 2-opt运算符只适用于边缘任务。虽然它可以很容易地修改来处理弧任务, 但这的工作在文献中仍然缺乏。

上述移动运营商首先被提出来解决车辆路线问题(VRP)[25], 然后扩展并广泛应用于CARP[1], [2], [8]-[12], [26]。他们采用了相当简单的方案来产生新的解决方案,

因此, 很可能会产生与当前的解决方案非常相似的新解决方案。直观地说, 我们可以说, 这些传统的移动操作符有“小”的步长, 因此只能在一个“小”的社区内搜索。然而, 在CARP具有较大的解空间或容量约束严格的情况下, 小步长的局部搜索操作符可能表现不佳。在前一种情况下, 传统的移动算符可能需要更长的时间来找到全局最优, i.e., 随着解决方案空间的扩大, 搜索过程将变得低效。在后一种情况下, 随着容量约束变得更紧密, 解决方案空间将变得更加坚固, 并包含更多的局部最优。因此, 解空间中的可行区域很可能被不可行区域所隔离。一个小步长的局部搜索很容易陷入局部最优中, 可能无法从一个可行的解决方案“跳转”到另一个解决方案。因此, 它可能永远不会适当地搜索所有可行的区域。

显然, 对于上述两种情况, 更希望使用具有较大搜索步长的本地搜索。乍一看, 似乎可以获得一个大的搜索步长, i.e., 我们可以简单地多次应用传统的移动操作符。这样的想法可以在[27]中找到, 其中Liang等人。解决了另一种类型的组合问题——削减库存问题。但是仔细看看CARP的传统移动操作符, 我们发现它们都定义了一个大小为 $O(n)$ 的邻域(n^2), 其中 n 为任务数。例如, 单个插入从 n 个任务中选择一个任务, 有 $n+m-1$ 可能插入任务的位置, 其中 m 是路由数。因此, 单个插入最多可以生成 $n(n+m-1)$ 不同的解。交换操作符需要从 n 个任务中选择两个任务, 并且存在 $n(n-1)/2$ 个不同的选择。类似的观察。因此, 连续应用 k 次的单个移动算子定义了一个大小为 $O(n^{2k})$ 的邻域, 随 k 呈指数增长。因此, 当 k 变大时, 禁止枚举所有可能的解决方案。这个问题的一个简单解决方案是随机抽取巨大社区的一部分。然而, 通常情况下, 解决方案空间中的某些区域比其他区域更有前途。因此, 随机抽样有点盲目, 可能会浪费大量的计算资源。综上所述, 虽然一个大的步长局部搜索可能是有益的, 但它不能通过简单地扩展传统的移动操作符来实现, 因此需要一种更精细的方法。为此, 我们开发了MS操作符。

III. 用于本地搜索的合并拆分操作符

MS操作符的目标是通过修改一个给定的解的多条路径来改进它。如其名称(MergeSplit)所示, 该操作符由两个组件 i 组成。e., 合并和拆分。给定一个解决方案, Merge组件随机选择它的 p ($p > 1$) 路由, 并将它们组合在一起, 形成一个无序的任务 id 列表, 其中包含所选路由的所有任务。拆分组件直接对“合并”生成的无序列表进行操作

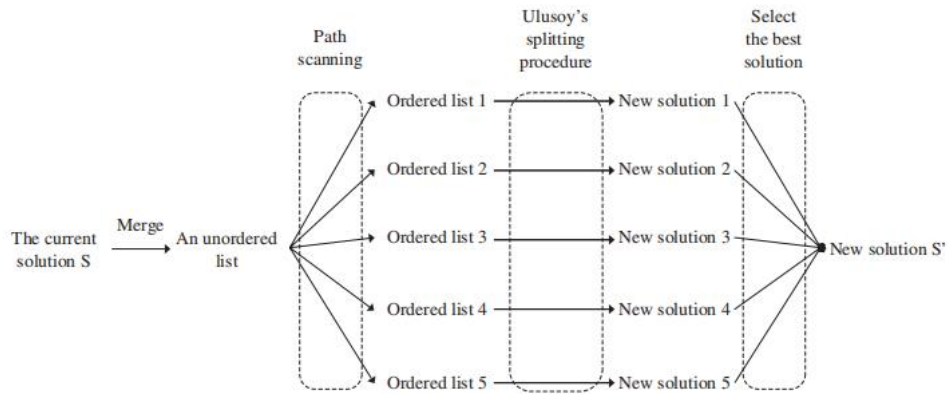


图5. 合并-Split operator.

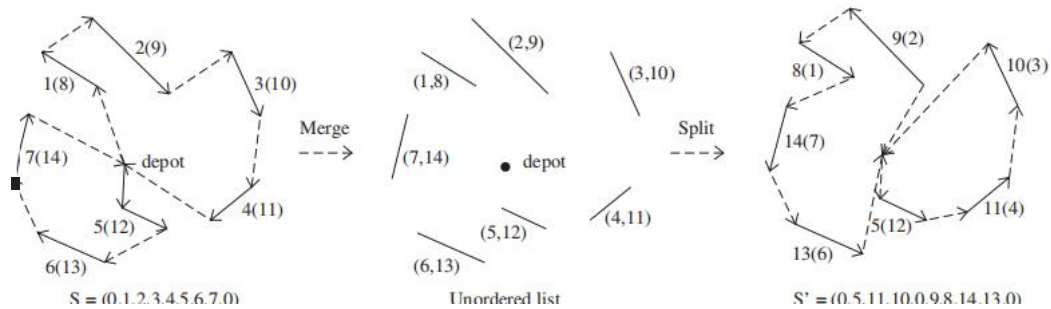


图6. 合并-拆分运算符的演示。

组成部分首先,应用路径扫描(PS)启发式[5]。PS首先要初始化一个空路径。在每次迭代中,PS都会找出不违反容量约束的任务。如果没有任务满足约束条件,则将当前路径的末端与仓库之间的最短路径连接起来,形成一条路径,然后初始化一个新的空路径。如果一个唯一的任务满足约束,PS将该任务连接到当前路径的末端(同样,它们之间的最短路径)。如果有多个任务满足约束条件,则选择最接近当前路径末端的一个任务。如果多个任务不仅满足容量约束,而且最接近当前路径的终点,则进一步采用五条规则来确定选择哪个:1)最大化任务负责人到仓库的距离;2)最小化任务负责人到仓库的距离;3)最大限度串联(t) ($sc(t)$),其中串联(t)和 $sc(t)$ 分别为任务 t 的需求和服务成本;4)减少串联(t)/ $sc(t)$;5)使用规则1)如果车辆小于停顿,否则使用规则2)。如果有多个任务仍然存在,那么这些关系就会被任意打破。在选择了无序列表中的所有任务后,PS将终止。请注意,PS并不会交替使用这五个规则。相反,它会扫描无序的任务列表五次。在每次扫描中,只使用一条规则。因此,PS将总共生成5个有序的任务列表。在分裂组件中,PS之后是Ulusoy的分裂过程[7]。换句话说,将Ulusoy的分裂方法应用于PS得到的所有五个有序列表,以进一步改进它们。给定一个有序的任务列表,Ulusoy的分裂过程是一种寻找最优方法的精确算法

将已排序的列表划分为不同的路由。由于Ulusoy的分裂过程是一种精确的算法,并且已经为人所知多年,因此我们在本文中省略了它的详细步骤。有兴趣的读者可参考原出版物[7]。

综上所述,MS操作符首先合并多条路由以获得一个无序的任务列表,然后使用PS对无序列列表进行排序。然后,利用Ulusoy的分裂过程,以最优的方式将有序列表分割成新的路由。最后,通过将新的路由嵌入到原解中,可以得到5个新的鲤鱼解,并选择最好的一条作为质谱算子的输出。图5演示了MS操作符的整个过程。

MS操作符的一个主要优点是它能够生成与当前解决方案有显著不同的新解决方案。如图所示。6、MS操作符生成一个解决方案 S' 基于解决方案 $S = (0, 1, 2, 3, 4, 1, 0, 5, 4, 0, 5, 6, 7, 0)$ 的 $S' = (0, 5, 11, 10, 0, 9, 8, 14, 13, 0)$ 。如果使用传统的移动操作符, S' 可以通过连续应用单次插入和双次插入来实现,但不能通过只应用任何一种传统的移动操作符一次来实现。因此,我们可以说MS操作符比传统的移动运营商具有更大的搜索步长。一般来说, p 越大。 $e.$,新的解决方案与当前的解决方案距离越远。MS运营商的另一个吸引人的特性是,它很可能产生高质量的新解决方案。这是由于采用了PS和Ulusoy的分裂程序,这两种方法都能够产生相对较好的解决方案。MS运算符的主要缺点是它的

```

输入：鲤鱼实例，psize，可大小，ubtrial， $PIs$ 
输出：可行的解决方案 $Sbf$ 
//初始化：
1. 设置当前人口 $pop = \emptyset$ ；
2 而  $|pop| < psize$  do
3 设置试验柜台编号=0；
4 重复
5 生成一个初始的解决方案 $Sinit$ ；
6  $ntrial \leftarrow ntrial + 1$ ；
7 至 西尼特不是任何解决方案 $\in$ 流行或=的克隆；
8 如果 西尼特是一个克隆的一些 $S \in$ 流行音乐，然后
9 打破；|
10 端
11 个  $pop \leftarrow pop \cup Sinit$ ；
12 端
13  $psize = |pop|$ ；
//主回路：
14. 当不满足停止标准时，请去做
15 设置一个中间人口 $popt = pop$ ；
16 为  $i = 1 \rightarrow$  可尺寸
17. 随机选择两种不同的解决方案 $S1$ 和 $S2$ 作为来自流行音乐的父母；
18 应用交叉操作符到 $S1$ 和 $S2$ 生成 $Sx$ ；
19 从0到1之间的均匀分布中抽取一个随机数 $r$ ；
20 如果  $r < PIs$  那时
21 应用本地搜索到 $Sx$ 生成 $S1s$ ；
22 如果  $S1s$  难道不是任何 $S \in popt$ 的克隆吗
23 是
24 其他的 如果  $Sx$  是  $nbtacib$  没有任何 $S \in$ 窗口
25 是
26  $popt = popt \cup Sx$ 
27 结束
28. 如果  $包$  不是一个克隆的任何 $S \in popt$ ，然后30 $\in$ =
29  $popt \cup 包$ 
30 端
31 端
32 端
33 使用随机排序方法对 $popt$ 中的解进行排序；
34 设置 $pop = \{最好的psize解决方案\}$ ；
35 端
36. 返回最佳可行的解决方案 $Sbf$ 在当铺里

```

图7. MAENS的伪代码。

与传统的移动算子相比，计算复杂度较高。幸运的是，通过MS操作员和其他搜索操作员的仔细协调，这种缺点可能或多或少地减轻这种缺点。在下一节中，我们将提出MAENS，其中MS和传统的移动算子被集成，形成局部搜索算子。

增值基于鲤鱼扩展邻域搜索的模因算法

在本节中，我们首先简要地总结一下MAENS。然后将描述算法的细节，包括初始化、交叉和局部搜索。

与大多数其他MAs一样，MAENS首先初始化解决方案开始。在每次迭代中，进行交叉和局部搜索来生成新的候选解， i 。e.，后代的人口。然后，利用随机排序[28]对亲本群体和后代群体进行组合，并对个体进行排序，这是将EAs应用于约束优化问题的常用方法。相同的解，也称为克隆，不允许同时出现在一个种群中，以保持种群的多样性。这条规则适用于初始种群和搜索期间的所有中间种群。MAENS的伪代码如图所示。7，其中 p 大小为种群大小， o 大小为每一代后代数量， $ubtiray$ 为最大数量

允许产生非克隆初始解的试验，以及 PIs 是对单个解进行局部搜索的概率。

1) 初始化：在初始化阶段，初始填充弹出被设置为空。然后生成非克隆的单个解决方案，并逐个插入到 pop 中。当已经生成了 $psize$ 非克隆解决方案或没有生成合格的解决方案时，初始化阶段终止。在后一种情况下， $psize$ 被重置为最终得到的解数。图中的第1到第13行。7演示了初始化阶段。

2) 交叉：在MAENS的每次迭代中，通过将基于序列的交叉（SBX）算子应用于从当前种群中随机选择的两个亲本个体，来实现交叉。每一对父母个体都会产生一个后代个体。SBX最初是为VRP [29]提出的，它被设计用于处理一个顶点 id 序列。然而，与CARP类似，VRP的目的也是按照最优顺序对序列中的元素进行排序，并将它们分成不同的组。因此，元素（ id ）是否表示图的顶点或边（弧）实际上并不重要，而SBX可以毫不费力地应用于CARP。给定两个父解决方案 $S1$ 和 $S2$ ，SBX随机选择两条路线 $R1$ 和 $R2$ 分别从他们。然后，两者 $R1$ 和 $R2$ 被进一步随机分成两个子分支，比如 $R1 = (R11, R12)$ 和 $R2 = (R21, R22)$ 。在那之后， a

表i
少女的参数总结

名称	描述	价值
<i>psize</i>	人口规模	30
<i>ubtrial</i>	生成初始解的最大试验次数	50
<i>opsize</i>	不每一代所产生的后代	$6*psize$
<i>Pls</i>	进行局部搜索的概率（突变）	0.2
<i>p</i>	合并-拆分操作符中涉及的路由数	2（第一组实验）
		2、3、4（第二组实验）
<i>冈比亚</i>	最大生成次数	500

表二
结果在gdb基准测试集的解决方案的成本方面。“最好的”和“平均的”代表着最好的人
以及从30次独立运行中得到的平均结果。“ns”表示成功运行的次数

名称	V	E	磅	地毯	VND	LMA	TSA2	平均	MAENS标准NS	最好的	TSAbest
1	12	22	316	316	316	316	316	316.0	0.030	316	316
2	12	26	339	339	339	339	339	339.0	0.030	339	339
3	12	22	275	275	275	275	275	275.0	0.030	275	275
4	11	19	287	287	287	287	287	287.0	0.030	287	287
5	13	26	377	377	377	377	377	377.0	0.030	377	377
6	12	22	298	298	298	298	298	298.0	0.030	298	298
7	12	22	325	325	325	325	325	325.0	0.030	325	325
8	27	46	348	352	350	350	348	348.7	1.020	348	348
9	27	51	303	317	315	303	303	303.0	0.029	303	303
10	12	25	275	275	275	275	275	275.0	0.030	275	275
11	22	45	395	395	395	395	395	395.0	0.030	395	395
12	13	23	458	458	458	458	458	458.0	0.030	458	458
13	10	28	536	544	544	536	540	536.0	0.030	536	536
14	7	21	100	100	100	100	100	100.0	0.030	100	100
15	7	21	58	58	58	58	58	58.0	0.030	58	58
16	8	28	127	127	127	127	127	127.0	0.030	127	127
17	8	28	91	91	91	91	91	91.0	0.030	91	91
18	9	36	164	164	164	164	164	164.0	0.030	164	164
19	8	11	55	55	55	55	55	55.0	0.030	55	55
20	11	22	121	121	121	121	121	121.0	0.030	121	121
21	11	33	156	156	156	156	156	156.0	0.030	156	156
22	11	44	200	200	200	200	200	200.0	0.030	200	200
23	11	55	233	235	235	233	235	233.0	0.030	233	233
平均	—	—	253.8	255.0	254.8	253.9	254.0	253.8	—29.5	253.8	253.8
没有最好的	—	—	—	19	19	22	21	21	—	23	23
APD	—	—	—	0.35	0.30	0.03	0.07	0.01	—	0.00	0.00

通过替换就获得了新的路线R12和R22. 最后，有些任务可能在新路由中出现多次，或者有些任务在*RI*不再在新的路线上

提供服务。在前一种情况下，重复的任务将从新的路由中删除。在后一种情况下，丢失的任务将被重新插入到新的

路由中。重新插入可能会导致额外的成本和违反容量限制。因此，每个缺失的任务都被重新插入到这样一个位置，即重新插入到任何其他位置都不会导致更低的额外成本和更小的容量约束违反。如果有多个位置满足这个条件，则将任意选择其中一个位置。

3) 局部搜索：在使用SBX生成后代种群后，每个后代将通过有概率的局部搜索得到进一步改进 PLs 。在上一节中，我们介绍了一些用于本地搜索的传统移动操作符，

并提出了新的质谱算子。现在的问题是，如何利用这些操作符来有效地进行本地搜索。如前所述，传统的移动操作符的搜索步长较小，而MS操作符的步长相对较大，并且通常随着其中所涉及的路由数量的增加而增加。许多关于EAs的论文表明，两种类型的搜索操作符都是最好的。一个小的步长算子很容易陷入局部最优，而当当前的解接近它时，一个大的步长可能会错过全局最优。因此，一个很自然的想法是首先在当前解决方案周围的一个小区域内进行搜索。当达到局部最优时，我们扩展搜索步长，试图从它中跳出来。如果成功，步长将再次减少，以充分利用新的局部区域。根据这个想法，我们开发了MAENS的本地搜索过程，如下所示。

表三
结果在0a1基准测试集的解决方案的成本方面。“最好的”和“平均的”代表着最好的人
以及从30次独立运行中得到的平均结果。“ns”表示成功运行的次数

名称	V	E	磅	地毯	VND	LMA	TSA2	平均	MAENS标准NS	最好的	TSAbest
1A	24	39	173	173	173	173	173	173.0	0.030	173	173
1B	24	39	173	173	173	173	173	173.0	0.030	173	173
1C	24	39	245	245	245	245	245	245.0	0.030	245	245
2A	24	34	227	227	227	227	227	227.0	0.030	227	227
2B	24	34	259	260	259	259	259	259.0	0.030	259	259
2C	24	34	457	494	457	457	457	457.2	1.129	457	457
3A	24	35	81	81	81	81	81	81.0	0.030	81	81
3B	24	35	87	87	87	87	87	87.0	0.030	87	87
3C	24	35	138	138	140	138	138	138.0	0.030	138	138
4A	41	69	400	400	400	400	400	400.0	0.030	400	400
4B	41	69	412	416	414	412	412	412.0	0.030	412	412
4C	41	69	428	453	428	428	428	431.1	3.114	428	428
4D	41	69	526	556	544	541	530	532.9	3.315	530	530
5A	34	65	423	423	423	423	423	423.0	0.030	423	423
5B	34	65	446	448	449	446	446	446.0	0.030	446	446
5C	34	65	473	476	474	474	474	474.0	0.030	474	474
5D	34	65	573	607	599	583	583	582.9	2.221	577	577
6A	31	50	223	223	223	223	223	223.0	0.030	223	223
6B	31	50	233	241	233	233	233	233.0	0.030	233	233
6C	31	50	317	329	325	317	317	317.0	0.030	317	317
7A	40	66	279	279	279	279	279	279.0	0.030	279	279
7B	40	66	283	283	283	283	283	283.0	0.030	283	283
7C	40	66	334	343	335	334	334	334.0	0.030	334	334
8A	30	63	386	386	386	386	386	386.0	0.030	386	386
8B	30	63	395	401	403	395	395	395.0	0.030	395	395
8C	30	63	518	533	543	527	529	525.9	1.729	521	521
9A	50	92	323	323	323	323	323	323.0	0.029	323	323
9B	50	92	326	329	326	326	326	326.0	0.030	326	326
9C	50	92	332	332	336	332	332	332.0	0.030	332	332
9D	50	92	385	409	399	391	391	391.0	0.030	391	391
10A	50	97	428	428	428	428	428	428.0	0.030	428	428
10B	50	97	436	436	436	436	436	436.0	0.029	436	436
10C	50	97	446	451	446	446	446	446.0	0.030	446	446
10D	50	97	525	544	538	530	530	533.6	1.50	531	528
平均	—	—	343.2	350.8	347.5	345.2	344.9	345.1	— 27.8	344.5	344.4
不最好的	—	—	—	16	22	30	32	26	—	33	34

APD	—	—	—	1.60	0.82	0.27	0.22	0.26	—	—	0.15	0.13
-----	---	---	---	------	------	------	------	------	---	---	------	-------------

给一个后代个体包由交叉生成，三个传统的移动操作符，i. e., 单插入、双插入和交换分别应用于个人，以进行最佳改进的局部搜索。也就是说，对于每一个移动操作符，它可以达到的所有解决方案包会被检查，只有在找到更好的解决方案时，当前的解决方案才会被更新。当当前的解决方案无法再得到改进时，此过程将终止。通过这种方法，我们将得到三个新的解，并选择最好的一个作为这个阶段的输出。由于该阶段在三个运算符定义的邻域内进行穷举搜索，因此其输出保证为局部最优。然后，将MS算子应用于该局部最优解，形成了局部搜索的第二阶段。假设当前的解由m条路由组成，并且MS操作符对其中的p进行工作，其中

p是MS运算符的预定义参数。然后，MS运算符最多可以生成 C_m^p 不同的解决方案。这个数字对于大m来说非常大。此外，使用MS生成一个单一的解决方案已经更多了比传统的移动运营商更耗时。因此，它可能是禁止列举所有的 C_m^p 时可能的解决方案 C_m^p 太大了。因此，我们限制了MS操作员在这个阶段最多生成100个解决方案，这是通过在 C_m^p 可能性(所有的解决方案都将被列举出来，如果是 $C_m^p \leq 100$)。同样，这个阶段是一个最好的改进过程。只有当MS操作符生成了一个更好的解决方案时，当前的解决方案才会被更新。如果MS算子能找到更好的解决方案，我们将以与第一阶段完全相同的方式应用三种传统的移动算子来利用新的局部区域。否则，第二阶段

表iv

结果在egl基准测试集的解决方案的成本方面。“最好的”和“平均的”代表着最好的人
以及从30次独立运行中得到的平均结果。“ns”表示成功运行的次数

名称	V	R	E	磅	LMA	TSA2	MAENS平 均	标准	ns	最好的	TSAbest
E1-A	77	51	98	3548	3548	3548	3548.0	0.0	30	3548	3548
E1-B	77	51	98	4498	4498	4533	4516.5	17.6	13	4498	4498
E1-C	77	51	98	5566	5595	5595	5601.6	9.9	20	5595	5595
E2-A	77	72	98	5018	5018	5018	5018.0	0.0	30	5018	5018
E2-B	77	72	98	6305	6340	6343	6341.4	12.0	6	6317	6317
E2-C	77	72	98	8243	8415	8347	8355.7	35.9	22	8335	8335
E3-A	77	87	98	5898	5898	5902	5898.8	2.9	27	5898	5898
E3-B	77	87	98	7704	7822	7816	7802.9	27.3	23	7775*	7777
E3-C	77	87	98	10163	10433	10309	10321.9	18.0	11	10292*	10305
E4-A	77	98	98	6408	6461	6473	6475.2	10.3	1	6456	6456
E4-B	77	98	98	8884	9021	9063	9023.0	18.7	17	8998*	9000
E4-C	77	98	98	11427	11779	11627	11645.8	46.7	9	11561*	11601
S1-A	140	75	190	5018	5018	5072	5039.8	35.9	20	5018	5018
S1-B	140	75	190	6384	6435	6388	6433.4	8.6	1	6388	6388
S1-C	140	75	190	8493	8518	8535	8518.3	1.5	28	8518	8518
S2-A	140	147	190	9824	9995	10038	9959.2	34.6	28	9895*	9956
S2-B	140	147	190	12968	13174	13178	13231.6	63.2	7	13147*	13165
S2-C	140	147	190	16353	16795	16505	16509.8	51.8	16	16430*	16505
S3-A	140	159	190	10143	10296	10451	10312.7	26.5	6	10257*	10260
S3-B	140	159	190	13616	14053	13981	13876.6	67.8	29	13749*	13807
S3-C	140	159	190	17100	17297	17346	17305.8	41.4	7	17207*	17234
S4-A	140	190	190	12143	12442	12462	12419.2	33.2	24	12341	12341
S4-B	140	190	190	16093	16531	16490	16441.2	38.1	28	16337*	16442
S4-C	140	190	190	20375	20832	20733	20767.2	74.6	8	20538*	20591
平均	-	-	-	9673.8	9842.3	9823	9806.8	-	17.1	9756.8	9773.9
不最好的	-	-	-	-	7	4	2	-	-	24	12
APD	-	-	-	-	1.38	1.3	1.14	-	-	0.70	0.84

结束这就完成了的本地搜索过程

少女。

除了上述描述外，关于本地搜索程序的两个问题仍有待进一步阐述。第一个问题是如何选择MS运算符的参数 p 。

一般来说， p 可以设置为任何大于1的整数。但回想一下， C_m^p 随着 p 的增加而迅速增加，因此在 C 中找到最好的机会 C_m^p 随机抽样的可能性会随着 p 的增加而减少，最终降低算子的性能。此外，更大的步长永远不能保证更好的解决方案。因此，我们推荐 $p = 2$ 作为默认选择。我们的实验结果将证明， $p = 2$ 比其他的要好得多值，如3和4。

最后一个必须解决的重要问题是如何评估一个解决方案，并确定它是否比当前的解决方案更好。特别是，由于CARP是一个约束优化问题，搜索过程不可避免地会遇到不可行的解。如本节开头所述，在每一代MAENS中，亲本和后代种群经过交叉和局部搜索后结合在一起。然后，应用随机排序[28]得到下一代的亲本群体。

让较低的等级表示更好的个体和随机性

通过一种类似于气泡排序的程序对个体进行排序。比较每一对相邻的个体和它们的排名

如果排名较高的人更好，就会被交换。如果两个被比较的个体都可行，则将仅根据适合度进行比较。否则，两个个体将根据他们的适合度或他们违反约束，以预定义的概率进行比较。虽然随机排序可以很容易地应用于局部搜索过程，但由于涉及到较高的计算成本，这是不合适的。对于每个个体，在局部搜索中使用随机排序涉及到一个时间复杂度为 $O(2n^2 \text{ Niter})$ ，其中 n 是任务数和 Niter 是传统的移动操作符的迭代次数。考虑到文献中研究的大多数CARP实例的 n 通常大于50，在局部搜索的每次迭代中对单个个体进行随机排序可能需要相当长的时间。因此，我们在这里采用一种简单得多的方法。也就是说，通过对其违反约束条件的程度及其成本的加权和来评估解决方案的质量，如(5)所示

$$f(S) = TC(S) + \lambda * TV(S) \quad (5)$$

式中， $TC(S)$ 为 S 的总成本， $TV(S)$ 为违规的总成本，可以通过将 S 中所有线路的违规情况相加来计算出来。 λ 是一个惩罚参数，它可以平衡了成本和违反之间的权衡。认为 $f(S)$ 值最小的解是通过局部搜索得到的最佳解。(5)的使用减少了

表v

博伦等基准集的结果。在解决方案的成本方面。“最好的”和“平均的”代表着从30次独立运行中获得的最佳结果和平均结果。“ns”表示成功运行的次数

名称	V	R	E	LB	GLS	TSA2	MAENS	TSAbest	平均性病NS最好			
C1	69	79	98	1590	1660	1660	1707.0	23.9	5	1660	1660	
C2	48	53	66	1095	1095	1095	1095.7	3.7	29	1095	1095	
C3	46	51	64	875	925	925	927.8	3.9	18	925	925	
C4	60	72	84	1285	1340	1340	1342.7	4.5	22	1340	1340	
C5	56	65	79	2410	2475	2470	2522.3	30.0	4	2470	2470	
C6	38	51	55	855	895	895	907.5	3.4	1	895	895	
C7	54	52	70	1735	1795	1795	1795.0	0.0	30	1795	1795	
C8	66	63	88	1640	1730	1730	1732.3	4.3	23	1730	1730	
C9	76	97	117	1775	1825	1830	1852.8	21.5	7	1820	1820	
C10	60	55	82	2190	2290	2270	2317.8	43.8	10	2270	2270	
C11	83	94	118	1725	1815	1815	1853.7	34.1	7	1815	1815	
C12	62	72	88	1510	1610	1610	1610.0	0.0	30	1610	1610	
C13	40	52	60	1050	1110	1110	1122.0	21.4	17	1110	1110	
C14	58	57	79	1620	1680	1680	1687.3	10.8	15	1680	1680	
C15	97	107	140	1765	1860	1865	1896.5	16.3	2	1860	1860	
C16	32	32	42	580	585	585	585.2	0.9	29	585	585	
C17	43	42	56	1590	1610	1610	1618.3	17.8	24	1610	1610	
C18	93	121	133	2315	2410	2415	2411.7	18.9	20	2390*	2410	
C19	62	61	84	1345	1395	1400	1425.7	19.1	1	1395	1395	
C20	45	53	64	665	665	665	668.5	6.7	21	665	665	
C21	60	76	84	1705	1725	1725	1725.2	0.9	29	1725	1725	
C22	56	43	76	1070	1070	1070	1070.0	0.0	30	1070	1070	
C23	78	92	109	1620	1690	1700	1724.3	30.8	8	1690	1700	
C24	77	84	115	1330	1360	1360	1368.5	6.2	7	1360	1360	
C25	37	38	50	905	905	905	907.0	7.6	28	905	905	
平均	—	—	—	1449.8	1500.8	1501.0	1515.0	—	16.7	1498.8	1500.0	
不最好的	—	—	—	—	21	20	3	—	—	25	23	
APD	—	—	—	—	3.30	3.32	4.19	—	—	3.21	3.27	

计算复杂度 $O(2nNiter)$ 从而减轻了局部搜索中所涉及的计算成本。正如在约束处理文献中所述，确定一个适当的 λ 值是一个重要的任务[30]。直观地说，由于TV(S)通常比TC(S)小得多，因此需要归一化才能使这两项在相同的幅度内，这样TV(S)就不会被忽略。此外， λ 应该随着当前解决方案的总成本的增加而减少，而随着总违反量的增加而增加，从而为不同的解决方案提供不同的偏差。基于这些考虑，我们为MAENS设计了一个自适应的 λ 。当对解决方案S进行本地搜索时， λ 首先被初始化为

$$\lambda = \frac{TC(S_{best})}{Q} * \left(\frac{TC(S_{best})}{TC(S)} + \frac{TV(S)}{Q} + 1 \right) \tag{6}$$

在哪里 *斯贝斯特*表示迄今为止找到的最佳可行的解决方案。在(6)中，TC(Sbest)/Q作为归一化因子。术语TC(Sbest)/TC(S)使 λ 随着S总成本的增加而降低，术语TV(S)/Q使 λ 随着S的约束而增加。术语“1”是为了确保足够大的 λ ，如果S是一个非常大成本的可行解决方案。在局部搜索过程中，如果连续5次迭代达到可行解，则 λ 值减半，如果连续5次迭代达到不可行解，则 λ 值增加一倍。

V. 实验研究

为了评价质谱算子和MAENS的有效性，我们进行了两组实验。在第一组算法中，我们全面地比较了MAENS和一些最先进的算法。然后，研究了质谱算子的影响。特别是，我们在四种场景下研究了MAENS的性能，即从MAENS中移除MS操作符，并将p设置为2、3和4。

A. 实验设置

所有实验都在四个CARP实例的基准测试集上进行，称为gdb集[31]、val集[32]、egl集[33]–[35]和贝尔伦斯等。的设置[10]。gdb集由DeArmon在[31]中生成，由23个实例组成。val集由Benavent等人生成。在[32]。它包含了基于10个不同的图的34个实例。通过改变车辆的容量，根据每个图生成不同的实例。egl集是由Eglese基于来自兰开夏郡[33]–[35]地区的一个冬季磨粉应用程序的数据生成的。它由基于两个图的24个实例组成，每个图都有一组不同的所需边和容量约束。由Beullens等人生成的测试集。在[10]是基于佛兰德斯的城际道路网。它还包括四个

表VI

博伦等基准集的结果。在解决方案的成本方面。“最好的”和“平均的”代表着

从30次独立运行中获得的最佳结果和平均结果。“ns”表示成功运行的次数

名称	V	R	E	磅	GLS	TSA2	MAENS平 均	标准	ns	最好的	TSAbest
D1	69	79	98	725	725	740	745.0	0.0	0	745	740
D2	48	53	66	480	480	480	480.0	0.0	30	480	480
D3	46	51	64	415	415	415	415.2	0.9	29	415	415
D4	60	72	84	615	615	615	616.0	3.8	28	615	615
D5	56	65	79	1040	1040	1040	1040.0	0.0	30	1040	1040
D6	38	51	55	485	485	485	493.0	15.6	23	485	485
D7	54	52	70	735	835	835	847.3	17.7	13	835	835
D8	66	63	88	615	685	685	704.2	15.5	7	685	685
D9	76	97	117	680	680	680	680.0	0.0	30	680	680
D10	60	55	82	900	910	910	910.0	0.0	30	910	910
D11	83	94	118	920	930	960	935.2	6.2	4	920*	940
D12	62	72	88	680	680	680	680.0	0.0	30	680	680
D13	40	52	60	690	690	695	691.0	2.0	24	690	690
D14	58	57	79	920	930	940	931.0	3.1	27	930	930
D15	97	107	140	910	910	950	919.0	3.1	3	910	950
D16	32	32	42	170	170	170	170.0	0.0	30	170	170
D17	43	42	56	675	675	675	675.0	0.0	30	675	675
D18	93	121	133	930	930	930	934.2	8.7	23	930	930
D19	62	61	84	650	680	690	680.0	0.0	30	680	680
D20	45	53	64	415	415	415	415.2	0.9	29	415	415
D21	60	76	84	695	805	825	834.2	18.8	0	810	815
D22	56	43	76	690	690	690	690.0	0.0	30	690	690
D23	78	92	109	715	735	735	748.2	8.4	3	735	735
D24	77	84	115	620	670	670	683.5	19.3	18	670	670
D25	37	38	50	410	410	410	410.0	0.0	30	410	410
意思	-	-	-	671.2	687.6	693.2	693.1	-	21.2	688.2	690.6
不最好的	-	-	-	-	24	18	10	-	-	23	21
APD	-	-	-	-	2.38	3.02	3.18	-	-	2.48	2.74

子集，即集合C、D、E和F，每个集合包含25个实例。集合D和F的实例分别与集合C和E的实例共享相同的网络，但车辆容量更大。我们的研究共使用了181个实例。虽然MAENS可以直接应用于混合鲤鱼而不做任何修改，但所有这些基准测试集都是无向鲤鱼，i.e.，所有的任务都是边缘的任务。众所周知，混合鲤鱼比无定向鲤鱼更具挑战性。然而，很少有（如果有的话）混合鲤鱼的基准测试实例是公开可用的。因此，我们不得不将我们的实验研究限制在无定向鲤鱼。

在整个实验过程中，MAENS都采用了相同的参数。当达到了预定义的代数时，该算法就会被终止。表一总结了实验中使用的MAENS的参数设置。所有实验均进行了30次独立运行，本文报告了所获得的最佳结果和平均结果。

B. 将MAENS与现有算法进行比较

在比较研究中，我们考虑了目前存在的五种算法：包括地毯[8]、VND [9]、GLS [10]、LMA [11]和TSA2 [12]。这些算法的结果直接从原始出版物中获得。尽管存在一些其他的鲤鱼算法，但它已经被证明是[12]

上述算法都是最先进的技术。对于我们研究过的每个CARP实例（除了val集的实例10D之外），至少有五种算法中的一种被报道来获得最著名的解决方案。因此，与这些算法进行比较将足以让我们进行评估。在gdb集和val集上，有地毯、VND、LMA和TSA2的结果。LMA和TSA仅应用于egl集。Beullens等人对GLS和TSA进行了比较。的集合。表二至表八列出了在所有4个测试集上的比较算法所获得的最终解的成本。对表中内容的简要描述如下。

- 1) 以|V|、|R|和|E|为标题的列分别表示顶点数、所需边和总边。由于所有的边都需要在gdb和val集中提供服务，所以在表II和表III中省略了|R|列。
- 2) 标题为LB的列给出了迄今为止发现的实例的下界，这是从[10]，[12]，[36]–[39]收集的。
- 3) 对于MAENS，标题为平均值和Std的列提供了在30次运行中计算出的平均结果和标准偏差。领导着NS的纵队站立着以获取MAENS成功运行的次数。在这里，对于每个实例，一次运行都被认为是成功的

表VII

博伦斯等人的基准集e的结果。在解决方案的成本方面。“最好的”和“平均的”代表着从30次独立运行中获得的最佳结果和平均结果。“ns”表示成功运行的次数

名称	V	R	E	LB	GLS	TSA2	MAENS	TSAbest	平均性病NS最好		
E1	73	85	105	1855	1940	1935	1967.8	35.0	3	1935	1935
E2	58	58	81	1580	1610	1610	1615.5	14.2	24	1610	1610
E3	46	47	61	750	750	750	752.0	4.1	24	750	750
E4	70	77	99	1580	1610	1615	1684.3	21.1	2	1610	1615
E5	68	61	94	2130	2170	2160	2228.7	49.0	3	2160	2160
E6	49	43	66	670	670	670	670.0	0.0	30	670	670
E7	73	50	94	1780	1900	1900	1900.0	0.0	30	1900	1900
E8	74	59	98	2080	2150	2155	2150.5	1.5	27	2150	2150
E9	93	103	141	2160	2250	2300	2327.7	38.2	1	2235*	2295
E10	56	49	76	1690	1690	1690	1691.5	5.7	28	1690	1690
E11	80	94	113	1810	1850	1855	1932.0	44.5	2	1850	1840
E12	74	67	103	1580	1710	1730	1764.3	17.9	0	1710	1705
E13	49	52	73	1300	1325	1325	1335.3	22.2	24	1325	1325
E14	53	55	72	1780	1810	1810	1817.0	10.9	20	1810	1810
E15	85	107	126	1555	1610	1610	1617.8	13.0	12	1595*	1610
E16	60	54	80	1785	1825	1825	1825.0	0.0	30	1825	1825
E17	38	36	50	1290	1290	1290	1294.3	6.3	18	1290	1290
E18	78	88	110	1600	1610	1610	1612.3	6.3	26	1610	1610
E19	77	66	103	1400	1435	1435	1437.0	3.1	20	1435	1435
E20	56	63	80	950	990	990	990.0	0.0	30	990	990
E21	57	72	82	1700	1705	1705	1755.5	22.9	2	1705	1705
E22	54	44	73	1155	1185	1185	1187.5	6.3	25	1185	1185
E23	93	89	130	1395	1430	1445	1469.0	13.1	0.0	1435	1435
E24	97	86	142	1695	1785	1785	1822.2	26.3	6	1785	1785
E25	26	28	35	655	655	655	655.0	0.0	30	655	655
平均	—	—	—	1517.0	1558.2	1561.6	1580.1	—	16.7	1556.6	1559.2
不是最好的	—	—	—	—	19	18	5	—	—	22	21
APD	—	—	—	—	2.51	2.69	3.77	—	—	2.41	2.54

只有当得到的解不比比较算法得到的解差时。最后，以“最佳”标题的列显示了30次运行中获得的最佳解决方案的成本。

- 4) 对于TSA2，[12]中报道了两种溶液。以TSA2为标题的列显示了TSA2对所有实例使用相同的标准参数设置获得的结果，而结果在以标题的列中给出TSAbest通过对每个实例的TSA2参数进行微调，得到最佳解。
- 5) 对于每个表，在底部都包含了另外三行。第一行表示在一个测试集中的所有实例上为每个算法计算的平均值。并计算了下界的平均值，以供参考。第二行总结了算法在比较方法中获得最佳解的实例数。第三行计算每个算法从下界的平均百分比偏差（APD）。
- 6) 在所有的表中，结果都以粗体突出显示，表示MAENS在比较方法中获得最佳解决方案的实例。用“*”表示的结果表明，MAENS已经获得了新的最著名的解（i.e.，以前从未找到过），为相应的实例。

7) 在[10]中，博伦斯等人的结果。的四个测试子集的成本。因此，我们在表V-VIII中以相同的形式呈现了MAENS的结果。

MAENS的疗效可以从两个角度进行评价。e.，它在30次独立运行中所取得的最好的和平均的性能。由于MAENS是一种随机算法，而大多数比较算法（LMA除外）都是确定性的，因此基于MAENS的最佳性能进行比较可能是不公平的。然而，这种比较确实提供了一些关于maens搜索能力的信息。考虑到“最佳”标题的列，可以发现MAENS在181个CARP实例中有175个获得了最佳解，明显优于比较算法。特别是，即使在对每个实例的参数进行了微调后，TSA2也在152个实例上执行了最好的参数，更不用说剩下的算法了。更重要的是，MAENS成功地在egl集的12个实例和博伦斯等人的基准集的4个实例上找到了新的最著名的解决方案。在APD方面，MAENS在egl集和Beullens等人的集C和E上均优于其他算法。在Beullens等人的gdb集和集合F上的APD最低，但仅略低于最佳算法

表VIII

伯伦等基准集的结果。在解决方案的成本方面。“最好的”和“平均的”代表着从30次独立运行中获得的最佳结果和平均结果。“ns”表示成功运行的次数

名称	V	R	E	磅	GLS	TSA2	MAENS平均	标准	ns	最好的	TSAbest
F1	73	85	105	1065	1065	1085	1071.0	7.9	13	1065	1070
F2	58	58	81	920	920	920	920.0	0.0	30	920	920
F3	46	47	61	400	400	400	400.0	0.0	30	400	400
F4	70	77	99	930	940	960	963.5	8.4	1	940	945
F5	68	61	94	1180	1180	1180	1180.3	1.3	28	1180	1180
F6	49	43	66	490	490	490	490.0	0.0	30	490	490
F7	73	50	94	1080	1080	1080	1090.7	24.5	24	1080	1080
F8	74	59	98	1135	1145	1145	1145.0	0.0	30	1145	1145
F9	93	103	141	1145	1145	1170	1197.8	27.6	2	1145	1155
F10	56	49	76	1010	1010	1010	1010.0	0.0	30	1010	1010
F11	80	94	113	1015	1015	1015	1037.5	16.3	7	1015	1015
F12	74	67	103	900	910	910	939.5	33.0	11	910	910
F13	49	52	73	835	835	835	835.0	0.0	30	835	835
F14	53	55	72	1025	1025	1035	1065.5	14.4	3	1025	1035
F15	85	107	126	945	945	990	951.7	9.9	20	945	965
F16	60	54	80	775	775	775	775.0	0.0	30	775	775
F17	38	36	50	605	605	630	605.0	0.0	30	605	605
F18	78	88	110	835	850	850	861.2	23.3	18	850	850
F19	77	66	103	685	725	740	725.0	0.0	30	725	725
F20	56	63	80	610	610	610	614.8	0.9	1	610	610
F21	57	72	82	905	905	905	905.0	0.0	30	905	905
F22	54	44	73	790	790	790	790.0	0.0	30	790	790
F23	93	89	130	705	725	730	736.3	14.3	12	725	730
F24	97	86	142	975	975	1010	1001.3	17.1	4	975	1010
F25	26	28	35	430	430	430	430.0	0.0	30	430	430
平均	—	—	—	855.6	859.8	867.8	869.6	—	20.2	859.8	863.4
不最好的	—	—	—	—	25	16	12	—	—	25	18
APD	—	—	—	—	0.54	1.44	1.55	—	—	0.54	0.91

表ix

测试集上比较算法的运行时（CPU秒）（maens平均超过30次运行）

测试集	地毯	VND	GLS	LMA	TSA2	男爵
<i>gdb</i>	4.9	0.5	—	2.6	1.7	6.3
<i>0a1</i>	34.6	2.1	—	21.6	14.1	68.1
<i>egl</i>	—	—	—	263.5	204.0	702.1
Beullens等人的C组	—	—	59.0	—	60.1	233.1
Beullens等人的集合D。	—	—	24.1	—	26.9	309.2
Beullens等人的集合E。	—	—	56.5	—	64.4	226.6
Beullens等人的F组。	—	—	14.8	—	29.5	235.0

在Beullens等人的val集和集D上。所有上述观察结果都证明，MAENS能够在广泛的不同测试问题实例中获得比最先进的算法更好的解决方案，尽管由于其随机性，这种能力可能无法得到保证。

根据其平均性能将MAENS与其他算法进行比较，可能会更公平。出于类似的原因，考虑TSAbest的结果也是不合适的，因为它会为单个实例调整参数。因此，只需要采用标准的TSA2。从表格中可以发现，MAENS的平均性能也很有竞争力。MAENSs的平均结果与最佳解决方案相当

就一切情况而论实际上，MAENS的平均性能仍然优于地毯、VND和LMA（如表II-IV所示）。此外，MAENS总是从gdb和val集中获得了大多数实例的最佳解。由于LMA是五种比较算法中唯一的基于总体的（也是唯一的随机的）算法，因此MAENS和LMA之间的比较可能是特别有趣的。不幸的是，在[11]中并没有明确说明报告的结果是在多次独立运行中取的平均值，还是只是获得的最佳结果。因此，做一个真正公平的比较是困难的。然而，考虑到MAENS的平均结果优于LMA报告的平均结果，我们可以得出MAENS在溶液质量方面优于此的结论。除了解决方案的质量之外，它也是如此

表x

AVERGE APD of MAENS在基准集上的“p”值不同

测试集	30次跑中最好的一次			平均参加了30次跑步				
	无MS	P =2为P, =3为 MS MS		P = 4为没有P = 2为P = 3为 MS MS MS MS				P =4用于 MS
<i>gdb</i>	0.00	0.00	0.00	0.00	0.16	0.01	0.01	0.00
<i>0a1</i>	0.23	0.15	0.14	0.20	0.56	0.26	0.33	0.38
<i>egl</i>	1.51	0.70	0.77	0.82	2.53	1.14	1.20	1.35
Beullens等人的集合C。	3.39	3.21	3.27	3.38	5.45	4.19	4.27	4.53
Beullens等人的集合D。	2.54	2.48	2.62	2.75	4.04	3.18	3.61	3.83
Beullens等人的集合E。	2.94	2.41	2.62	2.71	5.15	3.77	3.82	4.12
Beullens等人的F组。	0.71	0.54	0.65	0.79	2.49	1.55	1.82	2.08

我很自然地会问MAENS有多稳定。e.，MAENS表现良好的可能性有多大。这个问题可以通过计算30次运行中成功运行的次数来评估。在这里，只有当MAENS获得了所有比较算法中成本最低的解时，运行才被认为是成功的。如表中所示，所有基准测试集的MAENS的平均成功运行次数都大于15次。特别是，MAENS总是得到最好的解(i. e.，在181个实例中，有79个运行成功)。这种观察结果或多或少地证明了MAENS的稳定性。

然而，MAENS存在由MS操作符带来的高计算代价。在表九中，给出了每个比较算法在四个测试集上的平均运行时间。在我们的实验中，MAENS是用C语言编码的，并使用Intel (R) Xeon (R) E5335 2.00 GHz运行。由于所比较的算法是在不同的计算机上实现的，所以已经进行了标准化，以便对运行时进行公平的比较。也就是说，本文中给出的所有运行时都是通过将原始出版物中的运行时除以某些因素而得到的。具体来说，地毯和VND是在图形Indigo2 (195 MHz) 上实现的；因此，在[8]和[9]中呈现的运行时被除以10。[10]中的GLS结果是使用奔腾II 500 MHz获得的，因此我们将那里的运行时除以4。LMA是在奔腾III 1 GHz [11]上实现的，而TSA是在奔腾移动1.4 GHz [12]上实现的。因此，在相应的论文中给出的LMA和TSA的运行时间分别被除以2和10/7。需要注意的是，CPU时间之间的比较是具有指示性的，因为我们无法访问其他影响计算时间的信息，如操作系统、编译器、程序员的编码技能等。从表九可以看出，MAENS比所有被比较的算法都耗时得多。我们将在今后的工作中解决这个问题。下一节将讨论一些潜在的想法。

C. 进一步分析了合并分割操作符的效果与设置

如前所述，在这组实验中考虑了四种场景。为了研究MS算子对MAENS的贡献，我们将其从算法中删除，同时保持所有其他部分不变。此外，我们还将MS算子的参数p设置为3和4，以研究如何实现

它影响了MAENS的性能。然后将得到的三种算法在所有实例上运行30次，并与使用p = 2的MAENS进行比较。表X总结了结果，其中给出了得到的最佳结果和平均结果。为了简洁起见，我们只给出了每个基准测试集的平均APD。最佳结果用粗体突出显示。我们可以清楚地看到，MAENS在p = 2时表现最好，其性能普遍随着p的增加而恶化。另一方面，虽然将p设置为3或4会导致性能较差；他们仍然优于没有MS操作者的MA。例如，如果我们考虑平均性能，将MS算符与p = 3和4. 这两者都增强了所有基准测试集的结果。在总之，实验结果表明，MS算子确实对我们的算法至关重要，如果没有先验知识表明有其他选择，那么p = 2是一个相当好的默认选择。

最后，我们可以观察到，MS的加入导致MAENS在egl集上的改善最为显著，而在gdb和val集上则不那么明显。仔细观察CARP实例，可以发现egl集由相当多的具有挑战性的实例组成，它们有190个边缘任务。另一方面，gdb和val集的实例包含不超过100个边缘任务。这意味着，egl集的实例通常有更大的解空间。因此，MAENS在egl集上的成功似乎验证了我们的假设，即一个大的步长局部搜索算子，如MS算子，将有利于那些具有大解空间的问题。

VI . 结论和未来的工作

在本文中，我们在MA的框架内研究了CARP。首先，提出了一种新的局部搜索算子，即MS算子。与现有的本地搜索操作符只修改候选解决方案的一小部分不同，MS操作符可能会产生显著不同的解决方案。因此，MS操作符能够使用较大的步长进行搜索。因此，MS算符不太可能被困在局部最优状态中。其次，我们提出了MAENS算法。MAENS在其局部搜索过程中结合了传统的和MS操作符的优点，从而在开发和勘探之间实现了良好的权衡。基于我们全面的实验研究，可以得出两个主要结论。首先，MAENS是

能够获得比最先进的算法更好的解决方案。MAs是一种非常具有前途的鲤鱼研究方法, 值得更深入的研究。其次, 从MAENS中删除MS操作符会导致解决方案质量的显著下降, 因此MS操作符在MAENS的成功中起着重要作用。从MAENS和LMA之间的比较中也可以得出同样的结论, 因为LMA采用了与MAENS相同的框架和传统的局部搜索算子, 但不包含MS算子。

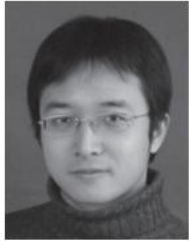
大搜索步长对于[40], [41]的数值优化已经得到了明确的证明, 大和小步长的良好组合可以更好地解决数值优化问题。质谱算子的成功表明, 类似的见解也适用于组合问题。值得注意的是, MS算符并不是专门针对MAENS提出的, 而是适用于任何现有的CARP算法。给定任何现有算法得到的候选解, MS算子可以应用于它。

虽然MAENS在我们的实验研究中表现出了出色的性能, 但它绝不是完美的。MAENS的主要缺点, i. e., 较高的计算成本, 仍有待于未来的解决。为了降低MAENS的时间复杂度, 可以考虑一种更精细的局部搜索方案。文献中普遍指出, 并非人口中的所有个体都值得在当地进行搜索。因此, 启发式可以包含在MAENS中, 只将局部搜索应用于那些“有希望的”(e. g., 低成本和可行的)个人。MAENS的另一个潜在改进可能是加入一个时变参数 p 。具体来说, 人们可以设计一个方案来监控MAENS的搜索过程, 并根据当前人口的状态来决定使用哪个 p 值。这样的方案可能会使MAENS更快地收敛到更好的解决方案。

参考文献

- [1] H. 汉达, D. 林, L. 查普曼和X. 姚, “利用进化算法进行盐析路线优化的稳健解决方案”, 姚教授。IEEE国会。Evol. 压缩。2006年, 温哥华, BC, 加拿大, pp. 3098 - 3105.
- [2] H. 汉达, L. 查普曼和X. 姚, “磨/盐卡车的稳健路线优化: CERCIA经验”, IEEE Comput. 知识Mag., 卷。1、没有。1, pp. 2月6日至9日。2006.
- [3] M. 错误, 弧路由, 理论, 解决方案和应用。马萨诸塞州波士顿: Kluwer, 2000年。
- [4] B. L. 金和R. T. 王, “带电电弧路由问题”, 网络, 第1卷。11日, 没有。3, pp. 305 - 315, 1981.
- [5] B. L. 金, J. S. DeArmon和E. K. “一类路由问题的计算实验”, 计算机。奥珀。物品, 卷。10, 没有。1, pp. 47 - 59, 1983.
- [6] H. I. 斯特恩和M. 错误, “路由抄表器”。奥珀。物品, 卷。6、没有。4, pp. 209 - 223, 1979.
- [7] G. “有能力的电路由的舰队规模和混合问题”, 欧盟。J. 奥珀。物品, 卷。22日, 没有。3, pp. 329 - 337, 1985.
- [8] A. 赫兹, G. 蒸发器和M. Mittaz, “有能力弧路由问题的标签搜索启发式”, Oper. 物品, 卷。48岁, 没有。1, pp. 129 - 135, 2000.
- [9] A. 赫兹和M. Mittaz, “无向能力弧路由问题的可变邻域下降算法”, 传输。科学., 卷。35岁, 没有。4, pp. 425 - 434, 2001.
- [10] P. Beullens, L. Muyldermans, D. 卡特莱斯和D. V. “有能力弧路由问题的引导局部搜索启发式”, 欧盟。J. 奥珀。物品, 卷。147, 没有。3, pp. 629 - 643, 2003.
- [11] P. Lacomme, C. Prins和W. “弧路由问题的竞争模因算法”, 安。奥珀。物品, 卷。131, 没有。1 - 4, pp. 159 - 185, 2004.
- [12] J. 白兰度和R. “有能力弧路由问题的确定性表搜索算法”, Comput. 奥珀。物品, 卷。35岁, 没有。4, pp. 1112 - 1126, 2008.
- [13] Y. Mei, K. 唐和X. 姚, “全球电路路径问题维修运营商”, IEEE运输。Syst. 人, 塞伯尔尼。B. Cybern., 卷。39日, 没有。3, pp. 723 - 734年6月。2009.
- [14] E. Ozcan和E. “并行代码优化的模因算法”, Int. J. 并行程序., 卷。35岁, 没有。1, pp. 33 - 61, 2006.
- [15] M. 唐和X. 姚, “超大规模集成电路地板规划的模因算法”, IEEE Trans. Syst. 人, 塞伯尔尼。B. Cybern., 卷。37日, 没有。1, pp. 62 - 69, 2月。2007.
- [16] Z. 朱, Y. S. Ong和M. Dash, “使用模包框架的包装过滤器特征选择算法”, IEEE Trans. Syst. 人, 塞伯尔尼。B. Cybern., 卷。37日, 没有。1, pp. 70 - 76年, 2月。2007.
- [17] X. 姚, “与扩展邻里的模拟退火”, Int. J. 压缩。数学, 卷。40岁, 没有。3, pp. 169 - 189, 1991.
- [18] E. W. “关于与图表有关的两个问题的注释”, 编号。数学, 卷。1、没有。1, pp. 269 - 271, 1959.
- [19] P. “关于进化, 搜索, 优化, 遗传算法和武术: 走向模因算法”, 加州理工大学并发计算机平台。加州理工学院, 帕萨迪纳, 加州, 项目代表。826, 1989.
- [20] R. 道金斯, 自私的基因。牛津大学。K.: 牛津大学, 1989年。
- [21] KH. - 梁, X. 姚和C. 牛顿, “近似n维景观的进化搜索”, Int. J. 基于知识的提示。雕刻西斯特., 卷。4、没有。3, pp. 172 - 183, 2000.
- [22] K. W. C. Ku, M. W. 麦克和W. C. “对循环神经网络的拉马克进化的研究”, IEEE反式。Evol. 压缩., 卷。4、没有。1, pp. 31 - 42, Apr. 2000.
- [23] Y. S. Ong和A. J. 基恩, “模因算法中的元拉马克学习”, IEEE Trans. Evol. 压缩., 卷。8、没有。2, pp. 99 - 110, Apr. 2004.
- [24] N. K. Bambha, S. S. 巴塔查里亚, J. Teich和E. “参数化局部搜索系统融入进化算法”, IEEE Trans. Evol. 压缩., 卷。8、没有。2, pp. 137 - 155, Apr. 2004.
- [25] M. Dror和L. “一种车辆路由改进算法的比较和库存路由的匹配实现”, Comput. 奥珀。物品, 卷。13日, 没有。1, pp. 33 - 45, 1986.
- [26] P. A. Mullaseril, M. Dror和J. 梁振英, “牲畜饲料分配中的分叉递送路径启发式方法”, J. 奥珀。物品社会, 卷。48岁, 没有。2, pp. 107 - 116, 1997.
- [27] KH. - 梁, X. 姚, C. S. 牛顿和D. 霍夫曼, “一种新的进化方法来减少股票问题。”奥珀。物品, 卷。29日, 没有。12, pp. 1641 - 1659, 2002.
- [28] T. P. 拉纳森和X. 姚, “约束进化优化的随机排序”, IEEE反式。Evol. 压缩., 卷。4、没有。3, pp. 284 - 294, Sep. 2000.
- [29] J. Y. Potvin和. “关于时间窗的车辆路径问题, 第二部分: 基因搜索”, 告诉J. 压缩., 卷。8、没有。2, pp. 165 - 172, 1996.
- [30] C. A. C. “与进化算法一起使用的理论和数值约束处理技术: 对艺术状态的调查”, 汇编。方法应用程序。机械。雕刻, 卷。191, 没有。11 - 12, pp. 1245 - 1287, 2002.
- [31] J. S. “对有能力中国邮差问题的启发式的比较”, M. S. 论文, 大学。马里兰州, 帕克学院, 马里兰州, 1981年。
- [32] E. Benavent, V. 坎波斯。科伯伦和E. “电路由问题。下界, “网络, 卷。22日, 没有。4, pp. 669 - 690, 1992.
- [33] R. W. 英语, “路线冬季磨车”, 离散Appl. 数学, 卷。48岁, 没有。3, pp. 231 - 244, 1994.
- [34] R. W. Eglese和L. Y. O. 李, “基于容量约束和时间期限的弧路由的启发式”, 《元启发式: 理论与应用》, 波士顿, 马: Kluwer, 1996, 页。633 - 650.
- [35] L. Y. O. 李和R. W. Eglese, “一种冬季加工车辆路径的交互式算法”, J. 奥珀。物品社会, 卷。47岁, 没有。2, pp. 217 - 228, 1996.
- [36] J. M. 贝伦格尔和E. “电路由问题的切割平面算法”, 计算机。奥珀。物品, 卷。30岁, 没有。5, pp. 705 - 728, 2003.
- [37] D. “对多重邮递员问题的贡献”, Ph. D. 论文, 鲁珀赫特-卡尔斯大学., 德国, 2004年。

- [38] R. 巴尔达奇和V. “基于无向弧路由问题的节点路由公式的精确方法”，网络，第1卷。47岁，没有。1，pp. 52 - 60，2006。
- [39] H. 长，D. A. M. Poggi和E. “使用对CVRP的转换来解决有能力的电弧路由问题，” Comput. 奥珀。物品，卷。33日，没有。6，pp. 1823 - 1837，2006。
- [40] X. 姚，Y. 刘和G. 林，“进化编程变得更快”，IEEE Trans. *Evol.* 压缩。，卷。3、没有。2，pp. 82 - 102，Jul. 1999。
- [41] C. Y. 李和X. 姚，“使用基于征税概率分布的突变的进化规划”，IEEE反式。 *Evol.* 压缩。，卷。8、没有。1，pp. 2月1日至13日。2004。



唐唐 (S05-07) 获得了B。雕刻2002年获中国华中科技大学武汉学位，获博士学位。D. 2007年毕业于新加坡南洋科技大学电气与电子工程学院。

他目前是中国科技大学计算机科学与技术学院自然启发计算与应用实验室的副教授。他是合著者。他的主要研究兴趣包括机器学习、模式分析、元启发式算法和现实世界的应用。国际期刊的编委会成员，IEEE大规模全球优化工作组



易梅 (S' 09) 收到了B元奖。S. 2005年在中国科技大学计算机科学与技术学院自然启发计算与应用实验室，合肥获得数学学位，目前正在攻读博士学位。D. 级别

他目前的研究兴趣包括模因子算法，路径搜索，和其他元启发式来解决电路路由问题。



辛耀 (M91-SM96-F03) 获得B。塞舌尔1982年获中国合肥市中国科技大学 (USTC) 硕士学位。塞舌尔1985年毕业于中国北京的华北计算技术学院，并获得博士学位。D. 1990年获得USTC学位。

从1985年到1990年，他在USTC担任副讲师和讲师，同时攻读博士学位。D. 模拟退火和进化算法。

1990年，他是一名博士后

澳大利亚堪培拉国立大学计算机科学实验室的海岸他的模拟退火和进化算法的工作。1991年，他在澳大

利亚堡小平建筑、建筑和工程部的联邦科学和工业研究组织的知识基础系统集团工作，在那里他主要从事一个关于污水管道自动检查的工业项目。1992年，他回到堪培拉，在澳大利亚悉尼的新南威尔士大学澳大利亚国防力量学院计算机科学学院担任讲师，后来晋升为高级讲师和副教授。自1999年4月以来，他一直担任美国伯明翰大学的计算机科学教授 (主席)。K. 他目前是美国伯明翰大学计算机科学学院计算智能和应用卓越研究中心的主任。K. 以及美国贸易大学计算机科学与技术学院自然启发计算与应用实验室的长江 (访问) 讲座教授 (张孔学者)。他在世界各地的会议和研讨会上发表了50多次被邀请的主题演讲和全体会议演讲。他有300多篇参考出版物。他的主要研究兴趣包括进化人工神经网络、机器学习系统的自动模块化、进化优化、约束处理技术、进化算法的计算时间复杂度、共同进化、迭代囚徒困境、数据挖掘和现实世界的应用。

资料暂存器2003年至2008年，姚担任IEEE关于EVOLU-EVOLU计算交易的主编，其他12种期刊的副编辑或编辑委员会成员，以及关于自然计算进展的世界科学书籍系列的编辑。他是中国科学院颁发的总统杰出论文奖。D. 模拟退火和演化方面的研究

1989年算法。他是2001年IEEE唐纳德G. 芬克奖授予他对进化神经网络的研究成果的论文奖。