

基于改进遗传算法的移动机器人动态路径规划

摘 要

本研究提出一种新型的遗传算法突变算子，并将其用于解决动态环境中的移动机器人路径规划问题。在有障碍物的环境中，移动机器人的路径规划是寻找一条从起始节点到目标节点的可行路径。遗传算法凭借其强大的优化能力，已被广泛用于生成最优路径。相较于简单的 GA 中的变异算子或者一些改进的变异算子可能导致不可行路径而言，本研究提出的突变算子不仅不会出现这种情况而且还可以避免过早的收敛。为了证明所提出方法的正确性，将其应用于两种不同的动态环境，并与以前文献提出的改进 GA 研究进行比较。采用所提出的突变算子的 GA 与其他方法相比，能够找到更多的最优路径并且更加快速的收敛。

1 简介

最近，随着技术的发展，研究人员对自动驾驶汽车的兴趣也在增加。文献中有许多关于自动驾驶汽车的研究。自动驾驶汽车研究的主要课题之一就是路径规划。路径规划会为移动机器人在有障碍物的环境中找到从起始节点移动到目标节点找到一条可行的路径。

路径规划既可以是静态的也可以是动态的。在静态环境中，在开始执行之前必须找到整个解决方案。但是对于动态或者部分可观测的环境而言，需要频繁地重新规划和更多的规划更新时间。一般来说，移动机器人路径规划的主要问题是计算的复杂性，局部最优解的存在性和适应性。研究人员在一直寻找可替代和更有效的方法来解决这些问题。

关于机器人路径规划的研究有很多方法，比如基于网格的 A 算法、道路图（Voronoi 图和可见图）、单元分解和人工势场等。以前的一些方法使用全局法来搜索工作空间中可能的路径，通常只处理静态环境，而且当环境复杂时计算成本很高。每种方法的有效性因应用环境的类型而异，都有自己的优缺点。例如，即使在静态环境中，人工势场方法也只能给出一个可能不是最短路径的解决方案路径。Ajmal Deena ALi 等研究了遗传算法(GAs)在机械手无碰撞路径规划的可行性，以减少搜索时间提高解的质量。结果表明该方法无论是在搜索距离还是计算时间上都优于传统的 A 搜索法。Chen 和 Zalzal 比较了 GA 和改进的 A 搜索法在移动机器人路径规划中的应用。观察到，尽管改进的 A 搜索法求解速度快于 GA，但是更容易陷入局部最小值，而且 GAs 总是可以求解出全局最优解或者接近全局最优解的解。

与传统的搜索和优化方法，如基于微算的和枚举策略相比，进化算法是鲁棒的，全局的，通常更容易适用于需要解决的问题很少或没有先验知识的情况。在路径规划方面优于其他技术的一种技术是遗传算法，因为它能够探索解空间，同

时保留已经找到的最佳解。在过去的十年中，GAs 利用其较强的优化能力，被广泛地用于生成最优路径。GAs 被认为是对复杂和不行为的目标函数最健壮地搜索技术之一。使遗传算法在开发接近最优解时具有吸引力的基本特征是，它们本质上是并行搜索技术。它们可以以并行的方式同时搜索所有的工作环境，因此它们可以更快地找到更好的解决方案。

近年来，许多基于遗传算法的路径规划算法都是通过制定遗传算子来实现的，其中大部分同时也是改进的方法。他们成功地找到了更好的解决方案。虽然传统的随机变异算子在简单遗传算法中会导致一个不可行路径，但本研究中提出的突变算子增加了种群的多样性，避免了过早收敛。

本文的组织结构如下：第二部分阐述了 GAs 在移动机器人路径规划中的应用，第三部分分析了本文提出的变异算子，第四部分对实验研究和结果进行评价，第五部分总结了本文的主要研究成果。

2 遗传算法路径规划

GA 是一种模拟自然遗传操作的并行全局搜索技术。因为它同时计算参数空间中的许多点，所以更有可能收敛到全局最优。搜索空间不一定是可微或者连续的。

2.1 环境的表示

许多路径规划方法都使用基于网格的模型来表示环境空间，如图 1 所示。基于网格的表示更容易计算距离和表示障碍物。基于网格的环境空间有两种方式表示，一种是有序编号的网格[1,13-15]，另一种是(x,y)坐标平面[9,16,17]。在文献中可以得出，有序编号的网格表示被广泛使用，因此在本研究中使用了这种表示方法。

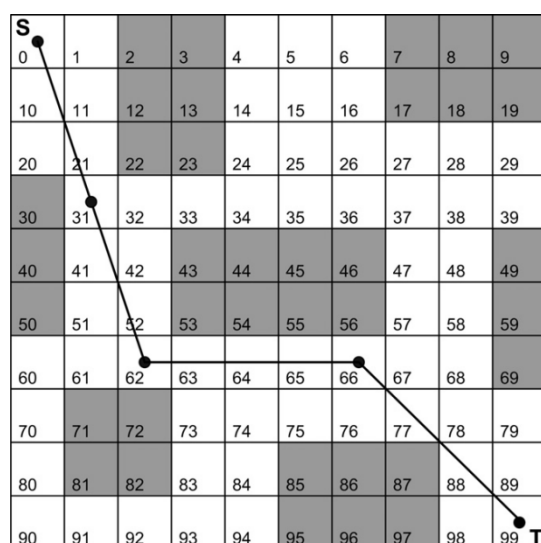


图 1 基于网格的环境表示法

2.2 染色体编码

染色体是路径规划问题的一个候选解决方案。染色体或者路径由起始节点、目标节点和移动机器人经过的节点组成。这些路径中的节点或者步骤被称为染色体的基因。不同的编码方法被用来创建染色体。一般使用二进制编码字符串方法

[9,18,19], 但是十进制编码的字符串方法也使用[1,13,14], 并且更灵活。十进制编码只需要更少的内存和优化空间。如图 2 所示, 一个有效路径由一系列的网格标签组成, 从起始节点开始, 到目标节点结束。

2.3 人口初始化

初始种群通常是随机生成的。一些染色体的产生可能包括与障碍物相交的不可行路径。即使初始种群包括不可行路径, 遗传算子也可以找到最优或者近似最优解。然而这个过程降低了算法的搜索能力, 增加了最优解的寻找时间。此外, 两个不可行染色体的交叉可能会产生新的不可行路径。为了解决这个问题, 在产生初始种群时, 必须检查每条染色体是否与障碍相交。如果确实如此, 染色体上的交叉基因就会被一个随机的可行基因改变。

在本研究中, GA 以随机生成和可行的初始人口运行。对于相同的环境和 10 次运行, 表 1 给出了每种方法的平均适应值, 生成次数和解决时间。如[13,14]所示, 一开始确定可行的初始人口对于 GA 的运行是非常有益的。

2.4、适应度函数

路径规划问题的目的是在起始节点个目标节点之间找到一条最优路径。最优路径可能是最短距离的、最少时间和能耗的路径。总之、在路径规划中, 目标函数被认为是最短路径。在本研究中。GA 使用的染色体目标函数值在以下方程中给出:

$$f = \begin{cases} \sum_{i=1}^{n-1} d(p_i, p_{i+1}), & \text{可行路径} \\ \sum_{i=1}^{n-1} d(p_i, p_{i+1}) + \text{惩罚}, & \text{不可行路径} \end{cases} \quad (1)$$

$$d(p_i, p_{i+1}) = \sqrt{(x_{(i+1)} - x_i)^2 + (y_{(i+1)} - y_i)^2} \quad (2)$$

这里 f 是适应度函数， p_i 是染色体的第 i 个基因， n 是染色体长度， d 是两个节点之间距离， x_i 和 y_i 是机器人当前的水平和垂直位置， $x_{(i+1)}$ 和 $y_{(i+1)}$ 是机器人的下一个水平和垂直位置。机器人的方向由以下公式决定：

$$\alpha = \tan^{-1} \frac{(y_{(i+1)} - y_i)}{(x_{(i+1)} - x_i)} \tag{3}$$

目标函数值定义为路径中个节点之间的距离总和。如果在机器人行进的方向上存在障碍物，则在目标函数上加上一个惩罚。惩罚值应该大于环境中的最大路径长度。为了寻找最优路径，该算法会寻找一个消除了惩罚的染色体。

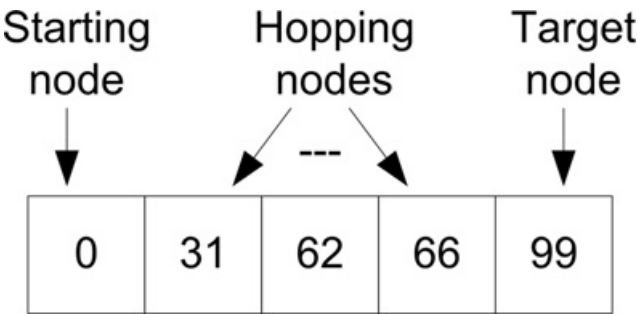


图 2 十进制编码染色体基因

表 1 初始种群方法的比较

初始人口方法	随机	可行性
适合度值	28.34	28.29
生成数	81	36
解决时间	0.87	0.49

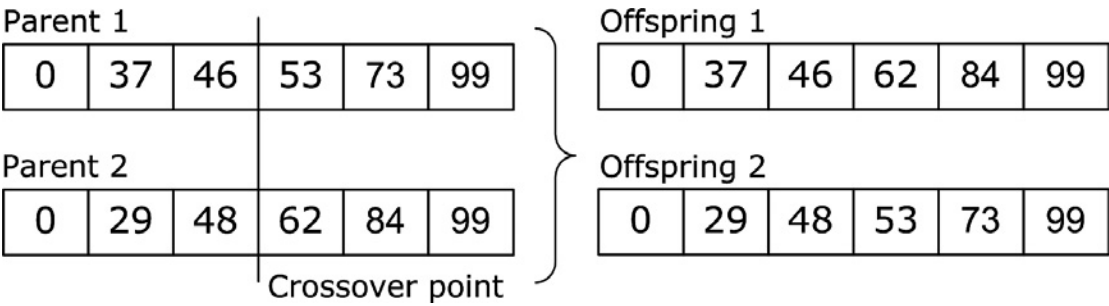


图 3 单点交叉

2.5 选择方法

遗传算法的主要原理是将染色体上的最优基因保存下来并遗传到下一代。在这一阶段，需要一个选择程序来决定最佳染色体。选择的过程包括三个步骤。第一步，找到所有染色体的目标函数值。第二步，将适应值根据目标函数值分配给染色体。在本研究中，使用基于秩的适应性分配来替代比例分配法。这可以防止一些较好的染色体在种群中占据主导地位。最后一步，根据适应值选择染色体，然后放入交配池产生新的染色体。

2.6 交叉算子

一般来说，交叉结合了双亲染色体的特征，形成两个后代。如图 3 所示，本研究使用的是单点交叉算子，交叉点之后两条染色体基因被交换。

2.7 突变算子

群体中的所有候选染色体在交叉操作之后都会经历随机突变。这是一种随机的按位二进制补码操作或基因的随机小变化，取决于染色体的编码，以突变率的概率统一应用于种群中所有个体的所有基因。突变算子将搜索空间扩展到了可能远离当前人群的区域，从而确保全局搜索。突变算子增加了种群的多样性，避免了过早的收敛。

3 一种新的路径规划突变算子

在传统的 GAs 中，随机突变是最常用的算子。但是随机突变会导致不可行路径的产生。如图 4a 所示，尽管染色体在突变操作之前是可行的，然而由于突变而产生的新节点可能存在障碍，因此会构成一条不可行路径。这会使得优化的速度变慢并增加了生成数的数量。

为了克服这个问题，文献中对改进突变操作进行了一些研究。其中最常见的是检查新突变染色体的可行性。如果不行，则在其上使用新的突变，知道产生新的可行突变。

Changan 等使用了一种改进的 GA，围绕现有的最佳个体周围寻找更好的个体，突变体被更好的个体取代。

Li 等人针对移动机器人路径规划问题提出了另一种更有效的变异改进方法。该方法从突变基因附近所有空闲节点的集合中随机选取一个节点，根据起始节点与目标节点的坐标所确定的前进方向接受该节点(图 4b)。如果变异操作后的新路径不可行或不理想，则重复随机节点选择，直到集合的搜索过程完成。

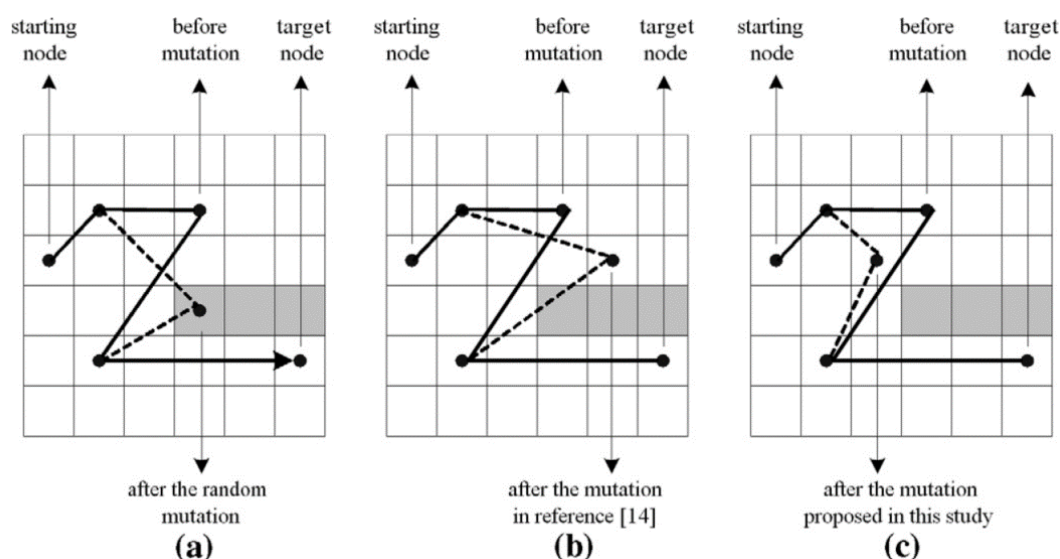


图 4 各种变异算子的比较

表 2 变异算子选取步骤

步骤	操作
1	从突变个体中随机选择一个非起始节点或目标节点作为突变基因
2	定义一个集合，由突变节点的所有可行(无障碍)相邻节点组成
3	确定所有路径的适应度值，每条路径由集合中的一个相邻节点组成
4	将适应度值最好的突变节点替换为原始突变节点

本文提出了一种新的变异算子方法，如表 2 所示。虽然看起来与 Li Q Zhang 提到的方法相似，但本文提出的方法与 Li Q Zhang 提到的方法有两个不同之处。首先，提出的突变方法是同时检查突变节点附近的所有空闲节点，而不是逐个随机选择节点;这意味着所提出的方法可以保证找到最佳节点，但在找到最佳节点之前，一个接一个的随机选择可能会找到更好的节点。第二，本文方法是依据总路径的适应度值而不是通过突变节点的移动方向接受该节点(如图 4c 所示)。最优路径可以由适应度值确定，即使突变节点的方向与起始节点到目标节点的方向相反。

例如，从图 1 所示的环境中选取一条由染色体(0,31,62,65,99)表示的路径。随机选取节点 65 作为突变基因。突变节点的相邻集合由节点{54、55、56、64、66、74、75、76}组成，其中节点 54、55、56 为障碍格。除了节点 54、55、56 之外，所有相邻的适应度值都可以找到。根据适应度值，将包含节点 65 的路径确定为距离最小的最适路径。这样，突变后的节点 65 将被节点 66 取代，新的染色体为 (0,31,62,66,99)。

4 动态环境实验与性能评估

为了证明该方法的成功，将其应用于两种不同的动态环境，并与文献中已有的改进遗传算法进行了比较。首先，处理 Li Q Zhang 中使用的环境。图 5a 为初始环境，即机器人开始移动前的环境，由 16×16 个网格组成，有 6 个障碍物区域(阴影区域)。图 5b 为原始环境的修改版本，即机器人开始移动后改变的环境，有 7 个障碍区域。GA 的参数说明如下。种群大小取 60，交叉概率取 1，变异概率取 0.3。

对于环境#1，分别才用随即突变法，Li Q Zhang 提出的突变法和本文提出的突变法逐一运行 GA。每种方法都进行 100 次遗传算法。

表 3 和 4 分别给出了初始环境和修改后环境的试验结果。表包含了最优解的数量，接进最优解的数量和不可行解的数量。表中还包含 100 次试验中最优和近似最优解的平均适应值、平均生成数和不可行解的数量。在表 3 和 4 中还可以清楚看到，使用本文提出的变异算子的 GA 分别找到了 54 次和 44 次最佳路径，而其他方法只找到了几次。该方法分别在 2 或 0 试验中没能找到可行路径，而其他方法在 5~29 次试验中均未成功。该方法的平均适应度值和平均生成数优于其他方法。但是即使所有方法都不到 1 秒，该方法的平均求解时间也比其他方法的求解时间差。

表 3 初始环境试验结果（图 5a）

	最优解	近似最优解	不可行解	适应度值	生成数	解决时间（s）
随机突变	0	46	54	26.69	81	1.22
Li 的突变	1	68	31	25.02	65	1.03
本文的突变	9	78	13	24.68	16	1.68

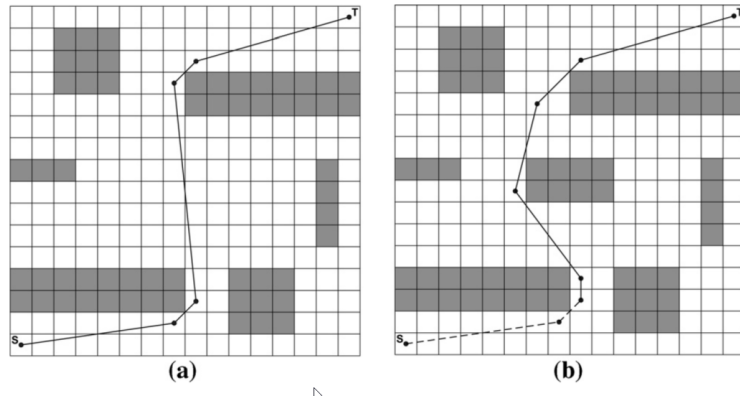


图 5 动态路径规划的初始 (a) 和修改 (b) 环境示例#1

表 4 改进环境的试验结果 (图 5b)

	最优解	近似最优解	不可行解	适应度值	生成数	解决时间 (s)
随机突变	0	95	5	35.37	23	0.20
Li 的突变	0	95	5	31.21	22	0.26
本文的突变	44	56	0	29.08	11	0.86

图 6 显示了所有方法的收敛性。采用该方法的遗传算法比其他方法收敛更快。因此，这使得该方法在动态环境中具有优势。

为了进行额外的比较，我们创建了一个比以前更复杂的新动态环境，如图 7 所示。图 7a 显示了初始环境，即机器人开始移动之前的环境，由 16×16 网格组成，有九个障碍区域（阴影区域）。图 7b 显示了原始环境的修改版本，即机器人开始移动后改变的环境，有十个障碍区域。遗传算法的参数如下。种群规模取 80，交叉概率取 1，变异概率取 0.2。对于环境#2，遗传算法采用与环境#1 相同的方法逐个运行。每种方法执行 GA 100 次。

表 5 和表 6 分别给出了初始和修改环境的实验结果。表格包含了在 100 次试验中发现的最优解的数量、接近最优解的数量和不可行解的数量。表中还包括 100 次试验中最佳和接近最佳解决方案的平均适合度值、平均生成数和平均解决时间。在表 5 和表 6 中可以清楚地看到，使用提出的变异算子的 GA 分别找到最佳路径 9 次和 32 次，而其他方法最多只能找到 2 次和 15 次。所提出的方法分别

在 13 次或 0 次试验中未成功找到可行路径，而其他方法分别在 31-59 次和 0-8 次试验中未成功。与其他方法相比，该方法的平均适应度值和平均生成数都更好。然而，尽管理所提出的方法的平均求解时间并不比其他方法的求解时间更短，但它们都很接近。

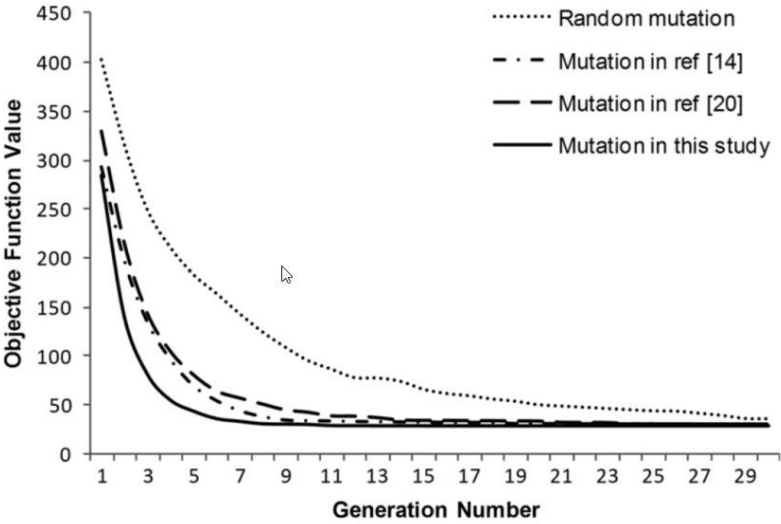


图 6 环境#1 下变异算子收敛性的比较

表 5 初始环境的实验结果（图 7a）

	最优解	近似最优解	不可行解	适应度值	生成数	解决时间（s）
随机突变	0	46	54	29.69	81	1.22
Li 的突变	1	68	31	25.02	65	1.03
本文的突变	9	78	13	24.68	16	1.68

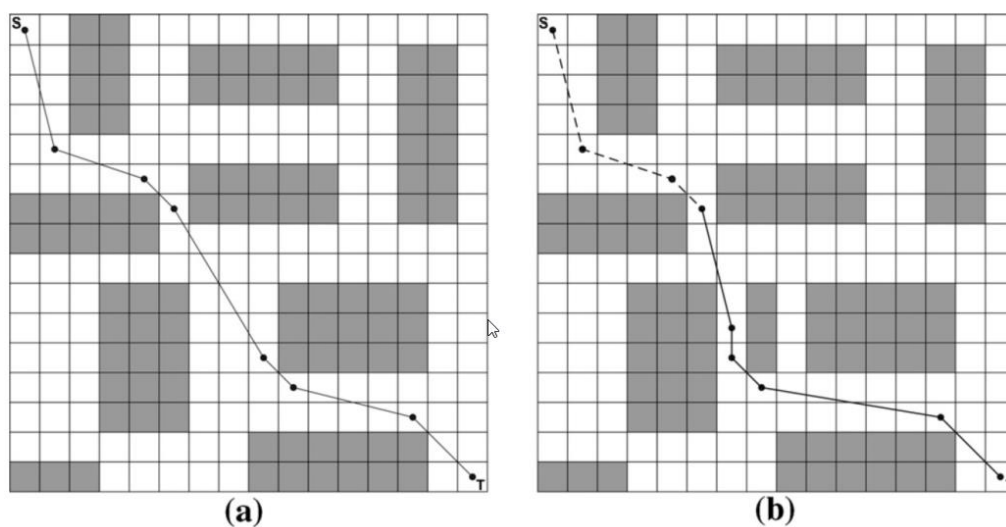


图 7 环境#2 下动态路径规划的初始 (a) 和修改 (b)

图 8 显示了所有方法的收敛性。采用该方法的遗传算法比其他方法收敛得更早。因此，这使得该方法在动态环境中具有优势。

表 6 改进环境的实验结果 (图 7a)

	最优解	近似最优解	不可行解	适应度值	生成数	解决时间 (s)
随机突变	6	94	0	25.48	73	0.74
Li Q 的突变	6	92	2	25.17	31	0.34
本文的突变	32	68	0	24.71	12	0.69

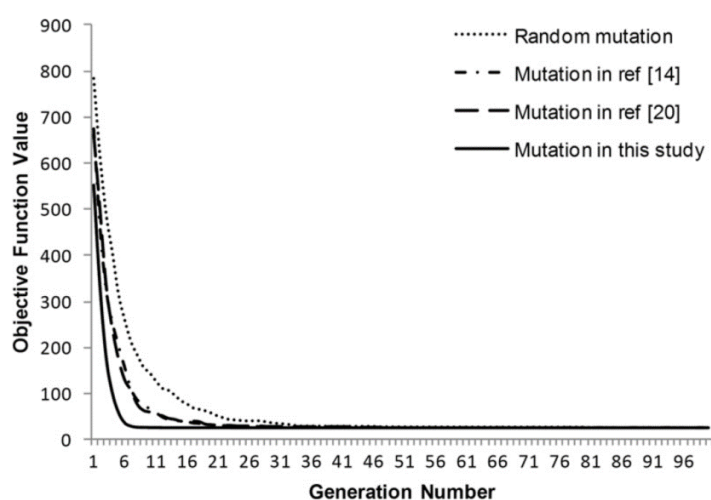


图 8 环境#2 下变异算子收敛性的比较。

结果明显表明,采用本文提出的变异方法的遗传算法比其他方法更频繁地找到最优路径。在所有四种方法中,本文提出方法的平均适应性值和平均生成数是最好的。此外,该方法具有最快的收敛速度。

5 结论

在本研究中，我们改进了一种新的遗传变异算子，并将其应用于移动机器人的路径规划问题。改进的变异方法不需要逐个随机选择节点，而是同时检查所有靠近变异节点的自由节点。该方法根据总路径的合适度值而不是通过变异节点的移动方向来接受节点。为了证明该方法的成功，将其应用于两种不同的动态环境，并与文献中先前的改进遗传算法研究进行了比较。从结果中可以清楚地看出，与其他方法相比，使用本文提出的变异算子的遗传算法可以找到更多的最优路径。该方法的平均适应度值和平均生成数优于其他方法。采用该方法的遗传算法比其他方法收敛更快。因此，这使得该方法在动态环境中具有优势。

致谢

这项工作得到了科凯利大学科学研究项目的支持，项目编号为 2010/97。