动态不确定环境下多目标路径规划方法

魏 唯 欧阳丹彤 吕 帅 冯宇轩
(吉林大学计算机科学与技术学院 长春 130012)
(吉林大学符号计算与知识工程教育部重点实验室 长春 130012)

摘 要 提出一种在动态不确定环境下求解多目标问题时快速调整移动路径的方法.首先提出采用逆向多目标启发式搜索进行全局规划,求解问题的最优路径集合;然后提出动态多目标路径规划方法,先根据当前观测进行全局规划,在移动过程中探测到不一致的环境信息时,通过对先前搜索中部分信息的重用,在全局规划的基础上进行增量重规划,调整当前状态与目标状态之间的移动路径.研究结果表明:采用增量重规划的求解方法通过搜索信息的重用提高求解效率,能够有效地处理动态不确定环境下的多目标路径规划问题.

关键词 多目标路径规划;动态不确定环境;启发式搜索;全局规划;增量重规划
中图法分类号 TP18 **DOI 号**: 10.3724/SP.J.1016.2011.00836

Multiobjective Path Planning under Dynamic Uncertain Environment

WEI Wei OUYANG Dan-Tong LU Shuai FENG Yu-Xuan
(*College of Computer Science and Technology, Jilin University, Changchun* 130012)
(*Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun* 130012)

Abstract In this paper, a method for adjusting the executing path of multiobjective path planning problems under dynamic uncertain environments is proposed. The backward multiobjective heuristic search algorithm is proposed firstly which starts the search process from the goal state to the start state to solve the given multiobjective problem. Then the dynamic multiobjective path planning algorithm is proposed. The algorithm performs global planning according to initial observations to the state space. Incremental replanning process is performed immediately when changes of environment are detected during the moving process. The replanning process can execute incrementally based on the global planning process to revise the path between any current state and the goal state efficiently by reusing parts of the saved information of the previous search. The experiment results show that the dynamic multiobjective path planning algorithm adopting an incremental replanning can solve multiobjective path planning problems under dynamic uncertain environments efficiently by reusing the information of previous search.

Keywords multiobjective path planning; dynamic uncertain environment; heuristic search; global planning; incremental replanning

1 引 言

路径规划是智能交通、通信网络、机器人等人工智能研究领域的重要分支.求解路径规划问题时,如果假定环境信息是静态完全可知的,可利用一次性的全局规划来获得一条初始状态到达目标状态的最优路径,但很多现实情况下,对于环境信息,特别是

收稿日期:2009-06-28;最终修改稿收到日期:2010-08-24.本课题得到国家自然科学基金(60773097,60873044,60873148,60973089)、符号计算与知识工程教育部重点实验室开放基金项目(93K-17-2009-K02,93K-17-2009-K06)和吉林大学研究生创新基金(20111060)资助.
魏 唯,女,1984 年生,博士研究生,主要研究方向为智能规划与自动推理. E-mail: weiwei.nenu@yahoo.com.cn. **欧阳丹彤**,女,1968 年生,教授,博士生导师,主要研究领域为基于模型的诊断与自动推理. E-mail: ouyangdantong@163.com. **吕 帅**,男,1981 年生,博士,讲师,主要研究方向为智能规划与自动推理. **冯宇轩**,男,1980 年生,博士研究生,主要研究方向为智能规划与自动推理.

动态障碍物的信息很难具有先验知识，实际移动过程中可能探测到真实情况与最初观测不同，这时只能根据实时探测到的环境信息，经过多次重规划来获得从当前状态到达目标状态的移动路径. 很多研究者已提出了不确定环境以及动态障碍物环境下求解路径规划问题的方法和策略[1-4]，其中增量搜索是一种非常有效的处理方法，对于不确定环境下的路径规划问题，能够对搜索现场所保留的信息进行有效地重用，从而减少工作量.

传统的路径规划只考虑单个衡量准则的优化. 然而，单个目标函数往往很难准确描述很多现实问题中多个相互冲突的衡量准则. 路径规划问题中需要优化的目标函数超过一个并需要同时处理时，称为多目标路径规划. 启发式搜索方法在人工智能的众多领域都有广泛应用，其用于解决多目标路径规划问题得到了相关研究者的关注. 多目标启发式搜索模型最早由 Stewart 与 White 提出，将 A^* 算法扩展到多个目标函数的情况，利用启发式搜索方法很好地处理了多目标路径规划问题[5]；Dasgupta 等人在搜索过程中使用非单调的启发信息，将多目标启发式搜索方法应用于 VLSI 设计问题[6]；Mandow 等人提出了路径扩展的方法，弥补了多目标启发式搜索中采用节点扩展方法的不足，提高了求解的准确性[7-8]；Dasgupta 与 Harikumar 等人提出了几种深度优先搜索方法用于解决多目标路径规划问题[6,9]. 目前，很多状态空间搜索规划器在求解智能规划问题时也考虑了多个衡量尺度的折中[10-12].

多目标的引入使得问题的求解与单目标条件下不同，由于各个目标之间通常都存在着冲突，一个解对于某个目标来说是较好的，对于其它目标来说可能是较差的，这样就造成多目标路径规划问题一般不存在唯一确定的最优路径，而是一组无法进行相互比较的最优路径的集合. 当移动过程中检测到变化的环境信息时，需要实时地重新规划出新的移动路径，这就要求以最快的速度计算出当前状态与目标状态之间的最优路径集合，因此，在动态不确定环境中进行有效的多目标路径规划增加了问题求解的难度.

文献[13]提出了一种多目标增量搜索算法，在求解过程中状态格局之间发生微小改变时，通过对先前搜索信息的重用，能够快速地得到新问题的最优解集，适用于求解一系列相似度较高的多目标问题. 本文进一步提出在移动过程中检测到环境信息的改变时，如何快速调整移动路线以保证全局工作最优性的方法，首先根据当前观测信息采用逆向启发式搜索方法进行全局规划，并选择一条实际移动路径，移动过程中探测到不一致的环境信息时执行局部重规划，重规划过程对先前搜索过程中的各项记录实时地更新，更新后在全局规划的基础上进行增量式搜索，求解当前状态与目标状态之间的最优路径集合. 实验结果表明：采用增量重规划的动态多目标路径规划方法通过搜索信息的重用与启发信息的引导提高了求解效率，能够有效地求解一系列动态不确定环境下的多目标路径规划问题.

2　相关定义

本节给出多目标路径规划问题的相关定义，下述定义中，向量表示路径的权重，向量的维数表示需要优化的目标函数的个数，每一维元素刻画一个单独的目标上的优化准则. 支配关系是向量之间的一种偏序关系.

多目标路径规划问题可以形式化地表示为三元组$\langle G, S_{\text{start}}, S_{\text{goal}}\rangle$，其中 $G=(N, A, \boldsymbol{c})$ 表示状态空间图，N 为状态节点集合，A 为弧的集合，$\boldsymbol{c}(m,n)$ 为转移代价函数，表示由状态节点 S_m 转移到状态节点 S_n 的 q 维代价向量，G 为无向图时，$\boldsymbol{c}(m,n)=\boldsymbol{c}(n,m)$；$S_{\text{start}}\in N$ 为初始状态节点；$S_{\text{goal}}\in N$ 为目标状态节点.

定义 1[6]. 设向量 $\boldsymbol{f}_1$ 和 $\boldsymbol{f}_2$ 为两个 K 维向量，称 $\boldsymbol{f}_1$ 弱支配 $\boldsymbol{f}_2$ 当且仅当 $\forall i, 1\leq i\leq K$ 满足 $f_{1i}\leq f_{2i}$，记作 $\boldsymbol{f}_1\leq\boldsymbol{f}_2$，其中 f_{1i} 和 f_{2i} 表示向量 $\boldsymbol{f}_1$ 和 $\boldsymbol{f}_2$ 的第 i 个元素.

定义 2[6]. 设向量 $\boldsymbol{f}_1$ 和 $\boldsymbol{f}_2$ 为两个 K 维向量，称 $\boldsymbol{f}_1$ 强支配 $\boldsymbol{f}_2$ 当且仅当 $\forall i, 1\leq i\leq K$ 满足 $f_{1i}\leq f_{2i}$ 且 $\boldsymbol{f}_1\neq\boldsymbol{f}_2$，记作 $\boldsymbol{f}_1\prec\boldsymbol{f}_2$，其中 f_{1i} 和 f_{2i} 表示向量 $\boldsymbol{f}_1$ 和 $\boldsymbol{f}_2$ 的第 i 个元素.

定义 3[6]. 对于向量集合 $\boldsymbol{X}$，称向量 $\boldsymbol{x}\in\boldsymbol{X}$ 为非支配的当且仅当 $\exists\boldsymbol{y}\in\boldsymbol{X}$ 使得 $\boldsymbol{y}\prec\boldsymbol{x}$，称 $nondom(\boldsymbol{X})=\{\boldsymbol{x}\in\boldsymbol{X}\mid\exists\boldsymbol{y}\in\boldsymbol{X}$ 使得 $\boldsymbol{y}\prec\boldsymbol{x}\}$ 为集合 $\boldsymbol{X}$ 中所有非支配向量的集合.

定义 4[7]. 设 P_{mn} 是状态节点 S_m 与状态节点 S_n 之间的所有路径的集合，路径 $p_1, p_2\in P_{mn}$，称 p_1 支配 p_2，当且仅当 $\boldsymbol{g}_1\prec\boldsymbol{g}_2$，记作 $p_1\prec p_2$，其中 $\boldsymbol{g}_1$ 与 $\boldsymbol{g}_2$ 分别是路径 p_1 与 p_2 的代价向量.

定义 5[7]. 设 P_{mn} 是状态节点 S_m 与状态节点 S_n 之间的所有路径的集合，路径 $p_1\in P_{mn}$，称 p_1 为非支配路径当且仅当 $\exists p_2\in P_{mn}$ 使得 $p_2\prec p_1$，即不存

在其它路径在所有目标上都优于路径 p_1.

指定初始状态 S_{start} 与目标状态 S_{goal}，多目标路径规划问题的最优解集即为 S_{start} 与 S_{goal} 之间的所有非支配路径的集合，它们的特点是无法在改进任何目标函数的同时不削弱至少一个其它目标函数，均视为一条最优路径.

上述定义了两种支配关系，原因如下：(1) 搜索过程中各个目标函数上权重都相等的两条路径具有同等的资格做进一步考虑，此时利用强支配关系比较所有待扩展候选路径；(2) 搜索过程中的一条路径的代价估值与已求出的一条解路径的权重相等时，该路径扩展出的任何路径必然是被这条解路径强支配的，已没有进一步扩展的必要，此时利用弱支配关系比较候选路径与解路径即可.

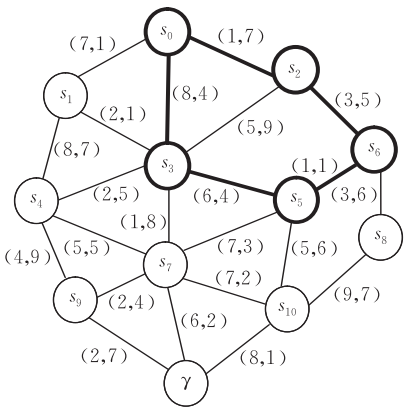
3 逆向多目标启发式搜索

当问题指定唯一确定的初始状态与目标状态，状态空间图为无向图时，可以采用逆向搜索方法，从目标状态开始执行搜索过程，求解初始状态与目标状态之间的最优路径集合.

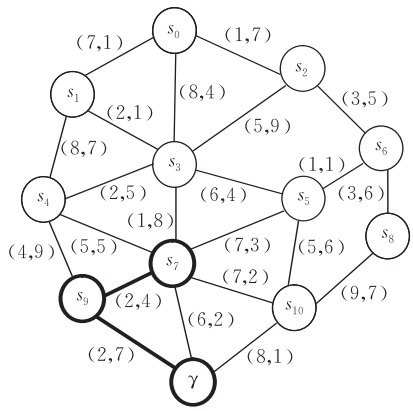
采用路径扩展方法的多目标启发式搜索方法每次扩展的基本单位是一条路径，利用路径的启发信息引导搜索过程以提高搜索效率.

定义 6[7]. 设 p 为搜索过程中到达状态节点 S_m 的一条路径，称 $\boldsymbol{f}(p)=\boldsymbol{g}(p)+\boldsymbol{h}_m$ 为路径 p 的代价估值，其中 $\boldsymbol{g}(p)$ 为路径 p 的代价向量，$\boldsymbol{h}_m$ 为 S_m 上的启发信息，估计每一个目标函数上从状态节点 S_m 到达目标状态的最小代价.

正向搜索时，路径 p 为从初始状态 S_{start} 到达状态节点 S_m 的一条路径，$\boldsymbol{g}(p)$ 为状态节点 S_m 的起始距离，$\boldsymbol{h}_m$ 为从 S_m 到达目标节点的代价估值. 逆向搜索时，p 为从目标节点 S_{goal} 到达状态节点 S_m 的一条路径，$\boldsymbol{g}(p)$ 为 S_m 的目标距离，$\boldsymbol{h}_m$ 为从 S_m 到达初始状态的代价估值. 如图 1 所示，其中(a)为正向搜索过程，当前状态为 s_5，$p_1=s_0\rightarrow s_2\rightarrow s_6\rightarrow s_5$ 为从初始状态 s_0 到 s_5 的一条路径，$\boldsymbol{g}(p_1)$ 为 p_1 的代价向量，$p_2=s_0\rightarrow s_3\rightarrow s_5$ 为初始状态与 s_5 之间的另一条路径，$\boldsymbol{g}(p_2)$ 为 p_2 的代价向量，$\boldsymbol{h}_5$ 为 s_5 与目标状态 γ 之间最优代价的估计，因此 $\boldsymbol{f}(p_1)=\boldsymbol{g}(p_1)+\boldsymbol{h}_5$ 为路径 p_1 的代价估值，而 $\boldsymbol{f}(p_2)=\boldsymbol{g}(p_2)+\boldsymbol{h}_5$ 为路径 p_2 的代价估值；(b)为逆向搜索过程，从目标状态 γ 开始执行，当前状态为 s_7，$p_3=\gamma\rightarrow s_9\rightarrow s_7$ 为目标状态与 s_7 之间的一条路径，$\boldsymbol{g}(p_3)$ 为从目标状态沿该路径到达状态 s_7 的代价估值，$\boldsymbol{h}_7$ 为状态 s_7 到初始状态 s_0 的最优代价估计，此时 $\boldsymbol{f}(p_3)=\boldsymbol{g}(p_3)+s_7$ 为路径 p_3 的代价估值. 可见，路径扩展方法对于到达同一个状态的各个路径进行区分，搜索过程用集合 $G_{op}(m)$ 记录到达状态 S_m 的未扩展的路径，集合 $G_{cl}(m)$ 记录到达状态 S_m 的已扩展的路径，因此，逆向搜索的扩展过程中 $G_{op}(m)\cup G_{cl}(m)$ 记录的是当前已找出的目标状态与 S_m 之间的所有非支配路径，表 $OPEN$ 记录当前各个状态节点的所有未扩展路径，每一个记录项为三元组 $(S_m,\boldsymbol{g}(p),\boldsymbol{f}(p))$ 表示的一条路径 p.



(a) 正向搜索细线



(b) 逆向搜索

图 1 正向搜索与逆向搜索

定义 7. 设 p_1 为表 $OPEN$ 中一条未扩展路径，$\boldsymbol{f}(p_1)$ 为路径 p_1 的代价估值，称 p_1 为一条非支配 $OPEN$ 路径当且仅当 $\exists(S_n,\boldsymbol{g}(p_2),\boldsymbol{f}(p_2))\in OPEN$ 使得 $\boldsymbol{f}(p_2)\prec\boldsymbol{f}(p_1)$.

多目标启发式搜索利用向量之间的支配关系从 $OPEN$ 表中选择将要扩展的路径并识别最优解路径，从非空的 $OPEN$ 表中选择一条非支配 $OPEN$ 路径进行扩展，并将扩展出的路径插入 $OPEN$ 表作为下一次扩展的候选路径，在此过程中，用 $COSTS$ 记录当前已求得的非支配解路径集合.

下面给出逆向多目标启发式搜索算法 BMHS

(Backward Muobjective Heuristic Search)的基本流程.

算法 1. BMHS($\langle G, S_{start}, S_{goal}\rangle$, $OPEN$, $COSTS$).

输入：多目标路径规划问题$\langle G, S_{start}, S_{goal}\rangle$，初始 $OPEN$ 表与 $COSTS$ 集合

输出：最优路径集合

1. 反复执行步 1.1～1.3，直到 $OPEN$ 表为空

1.1. 路径选择. 从 $OPEN$ 表中根据路径代价估值选择一条非支配 $OPEN$ 路径 p_x 进行扩展，令 $\boldsymbol{g}(p_x)$ 为路径 p_x 的代价向量，将 p_x 所对应的记录项 $(S_m, \boldsymbol{g}(p_x), \boldsymbol{f}(p_x))$ 从 $OPEN$ 表中删除，并将 p_x 从 $G_{op}(m)$ 移到 $G_{cl}(m)$；

1.2. 若 $S_m = S_{start}$，则 p_x 为一条解路径，$\boldsymbol{g}(p_x)$ 为解路径 p_x 的代价，将 $\boldsymbol{g}(p_x)$ 加入 $COSTS$ 并从 $OPEN$ 表中删除所有被 $\boldsymbol{g}(p_x)$ 支配的未扩展路径；

1.3. 若 $S_m \neq S_{start}$，则对 S_m 的所有前驱状态 S_n，生成到达 S_n 的路径 p_y，令 $\boldsymbol{g}(p_y) = \boldsymbol{g}(p_x) + \boldsymbol{c}(m, n)$，若 $\boldsymbol{g}(p_y)$ 不被 $G_{op}(n) \cup G_{cl}(n)$ 中任意路径代价支配，则

1.3.1. 将 p_y 插入 $G_{op}(n)$，并删除 $G_{op}(n) \cup G_{cl}(n)$ 中所有被 p_y 支配的路径；

1.3.2. 计算路径 p_y 的代价估值 $\boldsymbol{f}(p_y)$，若 $\boldsymbol{f}(p_y)$ 不被 $COSTS$ 中任何解路径支配，则将 $(S_n, \boldsymbol{g}(p_y), \boldsymbol{f}(p_y))$ 插入 $OPEN$ 表并记录相关的前驱信息；

2. 输出所求得的 S_{start} 与 S_{goal} 之间的最优路径集合.

算法 1 从输入接收 $OPEN$ 表初始值，对于给定的多目标路径规划问题，初始假设只有一条代价向量为零向量的路径到达目标节点，令 $OPEN = \{(S_{goal}, \boldsymbol{g}_{goal}, \boldsymbol{f}_{goal})\}$，$COSTS$ 为空. 每次循环首先判断 $OPEN$ 是否为空，当 $OPEN$ 为空时，搜索过程终止，输出所求得的最优路径集合；$OPEN$ 表非空时，从中选择一条非支配的 $OPEN$ 路径进行扩展.

对选出的一条路径 p_m 进行扩展时，生成到达状态 S_m 的所有前驱状态的路径，当生成的路径在所有到达该前驱状态的路径中非支配时，计算路径的代价估值，若生成的路径代价估值不受当前任何最优解路径支配，则将该路径插入 $OPEN$ 表；否则，该路径已没有继续扩展的必要，直接舍弃. 扩展到达初始状态时，则求得了初始状态与目标状态之间的一条解路径. 由于多目标路径规划问题需要求解一个最优解集，在求得一条解路径之后，搜索过程并不终止，而是反复扩展 $OPEN$ 表中其余路径，直到求出所有非支配解路径. 将新生成的路径 p_y 插入 $OPEN$ 表之前进行过滤，若 p_y 被 $COSTS$ 中的一条解路径支配，则 p_y 不被插入 $OPEN$ 表；每当求出一条非支配解路径时，过滤掉 $OPEN$ 表中所有被其支配的路径. 两次过滤过程保证了每次循环从 $OPEN$ 表中选出将要扩展的路径是不被任意解路径支配的.

上述多目标搜索过程反复执行直到求出所有非支配解路径，而单目标搜索过程，如经典的 A* 算法，只需要求出初始状态与目标状态之间一条最优路径，并且一旦得出最优路径，搜索过程立即结束. 如果目标函数的个数退化为单个目标，支配关系变成单纯的小于关系，那么算法 1 将变成求出所有最优路径的逆向 A* 算法，即若存在多条代价相等的最优路径，搜索过程将在逐个求出这些路径之后结束.

4　动态多目标路径规划

本节提出动态多目标路径规划算法 DMPP (Dynamic Multiobjective Path Planning)，执行过程分为全局规划与局部重规划. 全局规划获得初始状态与目标状态之间的最优路径集合，移动过程中探测到环境信息改变时，局部重规划过程重新计算出变化点与目标状态之间的移动路径，直到成功移动到目标状态.

4.1　动态多目标路径规划算法

当探测到实际情况与最初观测不符时，为避免停留时间过长，需要立即重新计算出当前状态与目标状态之间的最优路径. 本节提出的动态多目标路径规划算法进行局部重规划时，对受影响的记录进行更新，更新后在发生变化的状态与目标状态之间进行重规划，求得剩余的最优移动路径.

下面给出动态多目标路径规划算法 DMPP 的基本流程.

算法 2. DMPP(G, S_{start}, S_{goal}).

输入：多目标路径规划问题$\langle G, S_{start}, S_{goal}\rangle$

输出：到达目标状态的最优移动路径

1. 初始化. $OPEN = \{(S_{goal}, \boldsymbol{g}_{goal}, \boldsymbol{f}_{goal})\}$，$COSTS$ 为空集；

2. 全局规划. 执行逆向多目标搜索过程 BMHS($\langle G, S_{start}, S_{goal}\rangle$, $OPEN$, $COSTS$)，求得初始状态与目标状态之间的最优路径集合，从中选择一条实际移动路径；

3. 沿移动路径向目标状态移动，若成功到达目标状态，则输出所移动的路径，算法结束；

4. 若移动到 S_{change} 时探测到由 S_{change} 到达相邻状态 S_v 的情况发生改变，则执行局部重规划过程：

4.1. 执行更新过程 UPDATE(S_{change})；

4.2. 执行逆向多目标启发式搜索过程 BMHS($\langle G, S_{change}, S_{goal}\rangle$, $OPEN'$, $COSTS'$)，重新计算 S_{change} 与目标状态 S_{goal} 之间的最优路径集合；

5. 转步 3.

算法 3. UPDATE(S_{change}).

1. 更新 $G_{op}(S_{change}) \cup G_{cl}(S_{change})$ 中所有包括 $(S_{change}$,

S_v)的路径；

2. 令 $COSTS'=G_{op}(S_{change})\cup G_{cl}(S_{change})$，$G_{cl}(S_{change})=G_{op}(S_{change})\cup G_{cl}(S_{change})$，并将 $G_{op}(S_{change})$置为空集；

3. 对 S_{goal} 扩展到 S_{change} 过程中所有中间状态 S_u 的 $G_{op}(S_u)$集合中所有路径代价 $\boldsymbol{g}_u$，计算其到达 S_{change} 的代价估值 $\boldsymbol{f}_u$，若 $\boldsymbol{f}_u$不受$COSTS'$中任意解路径支配，则向 $OPEN'$插入记录(S_u，$\boldsymbol{g}_u$，$\boldsymbol{f}_u$).

算法 2 首先进行全局规划，求得 S_{start} 与 S_{goal} 之间的最优路径集合，从中选择实际的移动路径. 对于每个状态节点 S_m，$G_{op}(m)\cup G_{cl}(m)$记录了 S_m 与目标状态之间的所有非支配路径. 向目标状态移动的过程中实时探测环境信息是否发生改变，直到移动到目标状态，算法成功终止. 步 4 为探测到环境信息发生改变时的局部重规划过程. 当探测出 S_{change} 移动到 S_v的实际移动代价变化时，执行算法 3 的更新过程，对 $G_{op}(S_{change})\cup G_{cl}(S_{change})$中所有经过($S_{change}$，$S_v$)的路径代价进行修改；当 S_{change} 与 S_v 之间出现障碍物阻塞时，更新过程删除 $G_{op}(S_{change})\cup G_{cl}(S_{change})$中所有经过($S_{change}$，$S_v$)的路径代价. 更新后的 $G_{op}(S_{change})\cup G_{cl}(S_{change})$既为重规划过程所要求解的部分最优路径，遍历由 S_{goal} 扩展到 S_{change} 过程中所有中间状态 S_u的未扩展路径集合 $G_{op}(u)$，生成重规划搜索过程的初始 $OPEN$ 表. 从更新后的 $OPEN$ 表开始，仍以 S_{goal} 为初始状态，以 S_{change} 为新的目标状态执行逆向启发式搜索，重新计算 S_{change} 与 S_{goal} 之间的最优路径集合.

全局规划的搜索方向是从目标状态向初始状态，而重规划的搜索方向是从目标状态向变化状态，两次搜索可以看作是初始点相同但目标点不同的相似搜索过程，因此重规划能够重用全局规划所保留的各个状态的 G_{op} 与 G_{cl}，使得每一次重规划过程在先前搜索的基础上以增量的方式执行，先前的两部分信息能够被重用：(1) 初始 $OPEN$ 表直接获得各个状态节点的未扩展路径，而不是从空的 $OPEN$ 表经过一步步扩展逐个求出到达每个状态节点的路径；(2) 由 $G_{op}(S_{change})\cup G_{cl}(S_{change})$可以直接获得重规划过程的部分解路径，而不需要逐个求解.

4.2 算法可采纳性与复杂度分析

如果一个搜索算法对于任何存在解路径的图都能找到一条最优解路径，则该算法为 1 可采纳的；如果一个搜索算法对于任何存在解路径的图都能找到有限的最优解集，或者对无限的最优解集永不终止，则该算法是 N 可采纳的[7].

引理 1. 多目标启发式搜索的是 N 可采纳的[8].

定理 1. 动态多目标路径规划算法是 N 可采纳的.

证明. 根据引理 1，全局规划过程时可采纳的. 局部重规划过程首先对变化点 S_{change} 的 $G_{op}(S_{change})\cup G_{cl}(S_{change})$中所有受影响的路径进行更新，更新后的 $G_{op}(S_{change})\cup G_{cl}(S_{change})$为变化后 S_{change} 与 S_{goal} 之间当前已求出的所有非支配路径，作为重规划搜索过程的初始 $COSTS$ 集合，由此可以直接得到 S_{change} 与 S_{goal}之间部分解路径. 更新后对各个状态在全局规划中没有被扩展的路径计算到达 S_{change} 的代价估值，构成接下来搜索过程的初始 $OPEN$ 表. 由于 S_{change} 位置的变化对 S_{change} 与 S_{goal} 之间的其余状态的 $G_{op}\cup G_{cl}$没有任何影响，重规划的搜索过程能够直接重用这些信息，从更新后的 $OPEN$ 表与 $COSTS$ 开始，以 S_{goal} 为初始状态，S_{change} 为目标，执行逆向启发式搜索过程，继续求解 S_{change} 与 S_{goal} 之间的最优路径集合. 由此可见，重规划过程能够对先前全局规划所保留的部分信息进行正确地重用，因此，动态多目标路径规划算法是 N 可采纳的，启发信息 $\boldsymbol{h}_n\leq\boldsymbol{h}_n^*$ 时，算法始终能够求得移动过程中变化的状态与目标状态之间的最优路径集合. 证毕.

动态多目标启发式搜索算法的主要空间开销为全局规划对整个状态空间进行搜索的过程中存储到达各个状态节点的路径.

引理 2. 对于包含 L 个元素的 q 维向量集合，其中非支配向量个数平均为 $O((\ln L)^{q-1})$[14].

定理 2. 动态多目标路径规划算法最坏空间复杂度为 $O((\ln MN)^{q-1})$，其中 N 为状态空间中状态节点数，M 为到达各个状态节点的非支配路径个数的最大值.

证明. 引理 2 证明了对于元素个数为 L 的 q 维向量集合，其中非支配的向量的平均个数为 $O((\ln L)^{q-1})$. 在逆向多目标启发式搜索中，由于采用路径扩展方法，存储空间需要记录到达各个状态节点的所有路径，令 M 为到达一个状态节点的所有路径中非支配路径个数的上界，当状态空间中状态节点个数为 N 时，需要存储的路径个数最多为 MN，因此，q 个目标函数的动态多目标启发式搜索的最坏空间复杂度为 $O((\ln MN)^{q-1})$. 证毕.

搜索过程利用启发信息对路径到达目标状态的代价进行估计，能够更快地判断路径是否为一条非支配路径，当一条路径的代价估值 $\boldsymbol{f}$ 被一条解路径支配时，即 $\exists\boldsymbol{c}^*\in COSTS$ 使得 $\boldsymbol{c}^*\prec\boldsymbol{f}$，若启发信息 $\boldsymbol{h}_n\leq\boldsymbol{h}_n^*$，则 $\boldsymbol{f}\prec\boldsymbol{f}^*$，从而 $\boldsymbol{c}^*\prec\boldsymbol{f}\prec\boldsymbol{f}^*$，该路径到达目

标状态的实际代价一定是被支配的，因此该路径不需要继续扩展. 利用启发信息引导搜索过程能够提早判断被支配的路径，减少路径扩展数，提高求解的时间效率，重规划过程通过信息的重用，缩短了求解时间. 另外，算法的时间复杂度启发信息的定义与计算密切相关.

4.3　增量重规划与普通重规划的比较

全局规划过程中，采用正向搜索时，$G_{op}(m) \cup G_{cl}(m)$记录了 S_{start} 与 S_m 之间的所有非支配路径，搜索过程中记录了每个状态节点与 S_{start} 之间的非支配路径. 若采用逆向搜索进行全局规划，$G_{op}(m) \cup G_{cl}(m)$记录的是 S_{goal} 与 S_m 之间的所有非支配路径，搜索过程记录的是 S_{goal} 与每个状态节点之间的非支配路径. 当指定唯一确定的初始状态与目标状态时，全局规划过程采用正向搜索与逆向搜索的计算量是基本相同的.

局部重规划重新计算 S_{change} 与 S_{goal} 之间的最优路径集合. 全局规划采用正向搜索时，重规划过程要以 S_{change} 为新的初始状态，S_{goal} 为目标状态执行搜索，完全重新求解 S_{change} 与 S_{goal} 之间的多目标路径规划问题，搜索过程中求出的是各个状态节点与 S_{change} 之间的路径. 由于全局规划与局部重规划的搜索过程具有不同的初始状态，搜索过程中记录的路径代价不同，先前搜索对各个状态节点记录的路径信息无法被重规划过程重用，只能够重新执行一次搜索过程来对剩余的路径进行求解. 然而，若全局规划采用了逆向搜索方法，在重规划过程中将仍然以 S_{goal} 为初始状态，以 S_{change} 为新的目标状态执行搜索过程，由于前后两次搜索过程具有相同的初始状态 S_{goal}，不受变化影响的各个状态节点与 S_{goal} 之间的路径代价不变，因此全局规划与重规划过程中大部分状态节点的 $G_{op} \cup G_{cl}$ 集合中包含相同的路径信息. 重规划过程开始时，遍历各个状态节点当前的未扩展路径记录能够直接获得初始 $OPEN$ 表，而不是从空 $OPEN$ 表开始搜索过程，这将省略很大一部分扩展工作；另外，重规划的目的在于重新计算 S_{change} 与 S_{goal} 之间的最优路径集合，而全局规划的搜索方向是由 S_{goal} 向 S_{change} 进行的，$G_{op}(S_{change}) \cup G_{cl}(S_{change})$所记录的恰好是 S_{change} 与 S_{goal} 之间已存在的路径，这部分信息重用进一步减少了重规划搜索过程的计算量.

由此可见，增量重规划能够充分利用先前全局规划时每个状态节点所记录的信息，而不是彻底地重新求解，从而提高求解的效率.

5　实验测试与结果分析

本节采用二维 grid world 标准测试问题进行实验测试，用来做实验的机器配置如下：操作系统 fedora 8.0，CPU Intel Pentium 4 2.80GHz，内存 512MB DDR.

5.1　测试问题描述

4 个方向的 grid world 问题作为多目标路径规划问题的状态空间，初始位置为左上角的格子，目标位置为右下角的格子，用各个格子在 grid world 中的坐标来标识状态节点，每个状态节点的 4 个相邻格子作为其后继状态节点；向量的每一维元素值在[1,10]之间的整数中随机生成；对生成的多目标路径规划问题进行求解，选择经典的 Manhattan Distance作为每一个目标函数上的启发信息.

图 2 为动态不确定环境下多目标路径规划的问题描述，全局规划求得初始位置 S_{start} 与目标位置 S_{goal} 之间的最优路径集合，实际移动时从最优路径集合中选择满意程度最高的一条路径移动，图 2(a)为 S_{start} 与 S_{goal} 之间的最优路径集合. 图 2(b)由 S_{start} 开始的实线标记为实际移动路径，移动过程中探测环境信息，假设在 S_{change} 位置(灰色标记的格子)探测

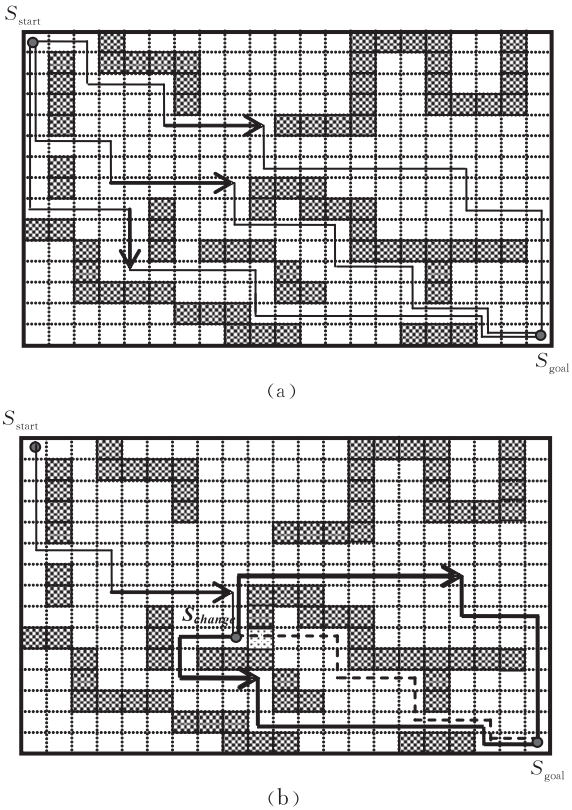


图 2　动态不确定环境下的路径规划问题

到环境改变，先前移动路径被阻塞，S_{change} 与 S_{goal} 之间的加粗实线标记为对剩余路径执行局部重规划需要求得的最优路径集合.

5.2 实验结果分析

测试分为 4 部分，由于实际问题中同时考虑两个衡量准则的情况最为常见，因此前 3 部分实验对两个目标函数的路径规划问题进行测试. 首先对生成的问题进行全局规划，由目标状态 S_{goal} 向初始状态 S_{start} 执行搜索，并从求得的最优路径集合中选择一条从 S_{start} 到达 S_{goal} 的移动路径. 第 1 部分实验测试移动过程中探测到动态障碍物的阻塞情况发生变化时增量重规划的求解效率，第 2 部分实验测试移动过程中探测到实际的移动代价发生变化时增量重规划的求解效率，第 3 部分对增量重规划过程中有无启发信息两种情况的结果进行比较，第 4 部分对目标函数的个数多于两个的问题进行测试，分析 DMPP 算法对不同目标函数的动态路径规划问题的求解能力.

第 1 部分实验中，格子可能被动态障碍物阻塞，当一个格子之间被阻塞时，4 个方向上相邻的格子无法通过它. 障碍物阻塞情况的动态变化将导致问题最优路径集合的改变. 当移动过程中探测到 S_{change} 周围格子的障碍物发生改变时，对探测到变化的位置 S_{change} 与 S_{goal} 之间的最优路径执行局部重规划. 选取移动路径上的不同位置进行测试，随机更改各个方向上障碍物的阻塞情况，表 1 为不同测试问题下采用增量重规划与普通重规划的测试结果比较.

表 1 动态障碍物环境下增量重规划与普通重规划的结果比较

	增量重规划结果				普通重规划结果		
Grid size=50×50	Initial OPEN size	Max OPEN size	Paths expanded	Time/s	Max OPEN size	Paths expanded	Time/s
prob01	218	350	1843	0.6718	478	2093	0.8145
prob02	172	870	2299	1.9406	1198	2437	2.4763
prob03	98	979	2783	2.2385	1386	3640	3.8145
prob04	125	1043	3477	4.1127	2467	5418	7.1218
prob05	73	2757	4642	6.1618	4219	7413	10.4074
Grid size=100×100	Initial OPEN size	Max OPEN size	Paths expanded	Time/s	Max OPEN size	Paths expanded	Time/s
prob01	329	761	3092	2.5706	985	3986	3.9584
prob02	436	1074	4879	5.4673	1211	5922	7.0238
prob03	208	1325	8594	16.8372	1982	16657	29.4732
prob04	199	2231	29894	44.2637	3359	44147	73.6754
prob05	78	2757	37723	70.2385	4219	71413	102.8967
Grid size=150×150	Initial OPEN size	Max OPEN size	Paths expanded	Time/s	Max OPEN size	Paths expanded	Time/s
prob01	943	1295	40293	77.5713	2238	54672	89.6574
prob02	1076	2088	88469	154.3219	3342	109834	191.2307
prob03	772	3864	110658	231.4539	4981	198736	306.5493
prob04	363	4873	189365	369.8340	6735	305439	587.9384
prob05	468	7659	246864	443.7482	8720	427763	739.5870
Grid size=200×200	Initial OPEN size	Max OPEN size	Paths expanded	Time/s	Max OPEN size	Paths expanded	Time/s
prob01	1843	7886	256844	539.5382	8003	298339	938.6543
prob02	893	7965	300928	795.3945	8320	409834	1543.9847
prob03	1009	8993	449384	1289.4905	8783	539845	2034.5392
prob04	983	9882	509438	1649.3937	9998	598473	2937.3829
prob05	694	12503	590334	2354.4220	11232	669484	4234.9833
Grid size=250×250	Initial OPEN size	Max OPEN size	Paths expanded	Time/s	Max OPEN size	Paths expanded	Time/s
prob01	1763	6795	332105	843.4324	6984	408797	1453.3720
prob02	2319	7886	398753	1124.3874	7978	435438	1993.2321
prob03	2008	8065	487968	1983.4524	9658	543002	3053.5393
prob04	1568	8976	527593	2847.3438	10928	609483	4048.3483
prob05	1098	13462	632065	3827.2438	14943	713204	5873.5430

表 1 中 Initial OPEN Size 为增量重规划由更新过程得到的初始 *OPEN* 表的大小，为重用全局规划的信息，Max OPEN Size 为搜索过程中 *OPEN* 表的最大值，体现了搜索过程用来存储未扩展路径的空间开销，Paths Expanded 为搜索过程的路径扩展数，表明搜索过程的计算量，Time/s 为局部重规划过程的执行时间. 对比两项实验结果，不难看出：增量重规划初始时能够直接获得部分 *OPEN* 路径，相

比于普通重规划，其 *OPEN* 表的最大值、路径扩展数与搜索时间均有改进，增量搜索过程存储 *OPEN* 表所需的空间更小，这是由于全局规划的搜索方向是从目标状态向初始状态执行，而重规划的搜索方向是从目标状态向变化的状态进行，这两个过程可以看作是初始节点相同但目标节点不同的两次相似的搜索过程，全局规划所记录的部分搜索信息对于重规划过程具有利用价值，因此增量重规划过程 *OPEN* 表中一些路径能够直接从全局规划过程继承，搜索过程所经历的路径数远远小于重新搜索的过程，另外，重规划过程开始之前已得到部分解路径，求出解路径的迭代次数更少，相比之下，普通的重规划过程不保留任何全局规划的信息，从空的 *OPEN* 表彻底重新求解所有解路径，求解效率较差. 当障碍物的动态改变使得重规划过程涉及到全局规划没有到达的区域时，将遇到一些全新的节点，节点信息的记录为零，这时增量重规划的整体求解时间有一些减慢，但大部分问题采用增量重规划时求解效率有所提高，一些问题的时间效率提高近 1 倍.

第 2 部分实验测试中，格子之间移动的代价随机发生改变. 初始时随机生成移动代价，移动过程中对路径上向前移动的代价进行随机更改，移动代价的改变不会阻塞先前的移动路径，但可能导致沿先前移动路径到达目标状态的代价不是最优的，此时执行局部重规划过程，重新求解变化后 S_{change} 与目标状态 S_{goal} 之间的最优路径集合，再从中确定将要移动的路径. 随机选取移动路径上的不同位置进行测试，对比不同的测试问题下采用增量重规划与普通重规划的实验结果，如表 2 所示.

表 2　移动代价可变环境下增量重规划与普通重规划的结果比较

	增量重规划结果				普通重规划结果		
Grid size=50×50	Initial OPEN size	Max OPEN size	Paths expanded	Time/s	Max OPEN size	Paths expanded	Time/s
prob01	65	564	1765	0.7638	883	1892	0.7769
prob02	137	785	1980	1.3402	936	2219	1.8763
prob03	81	886	2281	1.9863	1099	2392	2.7824
prob04	89	771	2163	2.0885	1030	3019	4.0082
prob05	73	2757	4642	7.8291	4219	7413	12.9201
Grid size=100×100	Initial OPEN size	Max OPEN size	Paths expanded	Time/s	Max OPEN size	Paths expanded	Time/s
prob01	132	837	2204	1.6492	759	2975	3.0392
prob02	392	930	3892	6.3975	876	4403	8.0385
prob03	190	1095	5043	11.0345	1092	14320	21.7859
prob04	309	2678	21743	34.5435	3921	35436	60.5436
prob05	128	4432	45396	89.5436	5094	79842	124.7689
Grid size=150×150	Initial OPEN size	Max OPEN size	Paths expanded	Time/s	Max OPEN size	Paths expanded	Time/s
prob01	583	902	30988	55.4930	1843	43025	79.8902
prob02	789	1893	75324	132.5439	2789	99432	200.8925
prob03	892	2490	94593	199.4367	3902	154369	325.4329
prob04	201	5436	204327	406.9879	8793	389543	690.8902
prob05	350	6782	278942	489.6782	7945	456367	823.6782
Grid size=200×200	Initial OPEN size	Max OPEN size	Paths expanded	Time/s	Max OPEN size	Paths expanded	Time/s
prob01	284	3782	109432	250.5439	5782	139015	478.6782
prob02	726	6782	278943	621.7892	9765	438902	1290.7896
prob03	1128	9023	399875	1094.7574	10943	589026	1928.8720
prob04	1463	16784	587809	2178.4583	19890	704329	3678.4794
prob05	974	14678	494320	1978.3567	20943	654320	3089.8652
Grid size=250×250	Initial OPEN size	Max OPEN size	Paths expanded	Time/s	Max OPEN size	Paths expanded	Time/s
prob01	678	4890	278953	638.4829	5693	397420	890.4392
prob02	902	6781	348302	902.4329	8021	489023	2090.8365
prob03	1903	7302	449824	1773.9264	10049	504320	3389.9265
prob04	1348	6792	408904	1532.8820	9902	589025	3890.3352
prob05	1796	8792	692389	4094.3392	17993	903214	6903.7991

由表 2 可见，大部分测试问题的增量重规划过程的 *OPEN* 表最大值，路径扩展数，搜索时间都远远小于普通重规划过程，更新过程中对只需对 $G_{\text{op}}(S_{\text{change}}) \cup G_{\text{cl}}(S_{\text{change}})$ 中部分路径代价进行更新，利用更新后的路径代价与其余路径进行支配关系的比较，移动代价变小时，搜索能够利用变化后的路径

代价快速地过滤掉其余所有被支配的路径，避免了不必要的扩展，移动代价变大时，先前的移动路径可能被其它路径支配，更新后生成重规划问题的初始 *OPEN* 表，由此开始执行新的搜索过程，减少了搜索过程中的大量计算与扩展工作. 更新过程生成 *OPEN* 表时需要对整个问题中未扩展的路径进行支配关系的比较，将所有不被当前任意解路径支配的路径插入 *OPEN* 表，占用了一些执行时间，并且当重规划的搜索过程扩展到全局规划没有到达的区域时，影响了重规划整体的时间效率，除个别问题时间效率提高不明显外，大部分问题时间效率有显著提高.

由上述实验结果可知，搜索信息的重用是提高增量重规划效率的关键，普通重规划无法重用全局规划中的任何信息，开始执行搜索过程时 *OPEN* 表为空，需要对重规划问题完全重新求解，效率较低. 当移动到接近目标状态的位置发生变化时，计算剩余移动路径的计算量较小，增量重规划与普通重规划的求解效率差别不大；若移动过程初期发生变化，即变化位置比较接近初始状态时，增量重规划的效率明显优于普通重规划，这是由于大部分移动路径需要重新确定，增量重规划能够直接获得部分解路径与 *OPEN* 表中的路径，避免了大量的重复扩展过程，节省了求解时间.

第 3 部分测试启发信息对求解过程的影响. 对各个规模的问题进行多次测试，表 3 记录的是 DMPP 算法求解过程采用启发式搜索与无启发信息搜索的平均路径扩展数与平均求解时间. 不使用启发信息的搜索过程对所有的路径都进行扩展与存储，而启发式搜索过程利用启发信息引导搜索过程，只对更有希望的路径进行扩展，对扩展出的路径利用支配关系进行过滤，只存储所有不被任何解路径支配的路径. 由表中实验结果可见，启发式搜索过程的路径扩展数与搜索时间都远远小于无启发信息的搜索过程，启发信息压缩了搜索空间，提高了求解的时间效率，部分问题的搜索时间提高约 1.5 倍. 从表 3 上看，问题规模越大，使用启发式信息相对于没有使用启发式信息的效率提升越不明显，这是由于多目标问题要求最终求解出所有的非支配解，并非只求解一个最优解，因此求解过程的计算量远远超出单目标问题，特别是随着问题规模的增大，所需计算量迅速增加. 启发信息用于提前识别非最优路径，加速求解，然而由于多目标问题本身的特点需要反复执行搜索过程，直到求出所有最优解，对求解效率也有一定的影响，无法单独凭借启发函数大幅度提高效率，但启发式搜索相对于无启发信息搜索的求解效率仍有明显的提高. 可见，使用启发信息引导搜索过程能够避免大量的扩展与计算过程，大大减少路径扩展数，在一定程度上缩短求解时间，特别是利用搜索方法求解多目标路径规划问题时，计算量较大，启发信息使得求解效率有显著的提高.

表 3 增量重规划过程中有无启发信息的搜索结果比较

格子大小	增量重规划搜索结果		无启发信息搜索结果	
	Paths expanded	Time/s	Paths expanded	Time/s
50×50	2999	5.1618	2846	7.7863
100×100	22348	42.4568	40894	61.1597
150×150	153481	300.4882	184874	476.7941
200×200	407458	1227.4589	454338	1534.7891
250×250	523989	3675.7846	556947	4054.9347

最后，进一步对目标函数的个数多于两个的路径规划问题进行测试. 对不同目标函数的情况分别选取 150×150 规模下多个的测试问题，表 4 所示为不同目标函数下增量重规划与普通重规划的平均路径扩展数与平均搜索时间. 实验结果表明对于各种不同目标函数情况下的动态路径规划问题，使用 DMPP 算法均能够很好地解决，求解效率也有不同程度的提高.

表 4 目标函数个数不同情况下的结果比较

	增量重规划结果		普通重规划结果	
	Paths expanded	Time/s	Paths expanded	Time/s
3-obj	108941	217.1887	169329	325.7291
4-obj	196723	386.4861	235483	493.9264
5-obj	187982	526.5697	298262	738.4392
6-obj	208392	590.4876	353724	802.7818
7-obj	239201	703.7633	438291	1038.3892
8-obj	332851	937.5468	568496	1331.6727
9-obj	457639	1157.5698	655945	1573.7293

上述实验结果表明动态多目标路径规划算法的优点：

(1) 局部重规划过程能够以增量的方式对先前搜索保留的信息进行有效重用，有两类信息能够被重用：① 全局规划过程的部分未扩展路径直接作为局部重规划过程的未扩展路径，记录在 *OPEN* 表中；② 全局规划过程的部分已扩展路径直接作为局部重规划过程的解路径，这两类信息的有效重用使得重规划节省了大量的计算时间.

(2) 利用启发式能够提前识别出非最优解，避免不必要的扩展，进一步上压缩搜索空间. 实验部分测试了利用搜索方法求解多目标问题时启发信息对求解效率的影响. 算法中的启发信息用于选择将要

扩展的路径以及对新生成的路径进行过滤. 路径选择步骤时，*OPEN* 表中的非支配 *OPEN* 路径被认为更有希望成为非支配解路径，将优先选择，每发现一条新的非支配解路径时，对 *OPEN* 表中所有被支配的路径进行过滤. 扩展路径时，对于新生成的路径，首先对路径代价进行过滤，再对路径的代价估值进行过滤. 可见，搜索过程中利用启发信息能够时刻检查路径之间的支配关系，对于没有希望的路径提早做出判断，减少搜索过程的计算量，启发信息越接近实际情况时，对路径实际代价的估计越准确，搜索空间的压缩比例越高.

5 总　结

本文提出了一种动态不确定环境下求解多目标路径规划问题的方法. 首先提出了采用逆向多目标启发式搜索进行全局规划，求解给定问题的最优路径集合；在此基础上，提出了动态多目标路径规划方法，该方法将多目标路径规划问题的求解过程分为全局规划与局部重规划，重规划以增量的方式对全局规划所保留的部分信息有效地重用，能够更快地调整变化位置与目标位置之间新的移动路径. 实验测试验证了动态多目标路径规划方法的求解效率，结果表明基于增量重规划的方法能够很好地处理一系列动态不确定环境下的多目标路径规划问题.

参　考　文　献

[1] Zhu Qing-Bao. Ants predictive algorithm for path planning of robot in a complex dynamic environment. Chinese Journal of Computers, 2005, 28(11): 1898-1906(in Chinese)
(朱庆保. 动态复杂环境下的机器人路径规划蚂蚁预测算法. 计算机学报, 2005, 28(11): 1898-1906)

[2] Koenig S, Likhachev M, Furcy D. Lifelong planning A*. Artificial Intelligence, 2004, 155(1-2): 93-146

[3] Koenig S, Likhachev M. D* Lite//Proceedings of the 18th National Conference on Artificial Intelligence and 14th Conference on Innovative Applications of Artificial Intelligence. MenIo Park: AAAI, 2002: 476-483

[4] Xi Yu-Geng, Zhang Chun-Gang. Rolling path planning of mobile robot in a kind of dynamic uncertain environment. Acta Automatica Sinica, 2002, 28(2): 161-175(in Chinese)
(席裕庚，张纯刚. 一类动态不确定环境下机器人的滚动路径规划. 自动化学报, 2002, 28(2): 161-175)

[5] Stewart S Bradley, White C. Chelsea. Multiobjective A*. Journal of the ACM, 1991, 38(4): 775-814

[6] Dasgupta P, Chakrabarti P P, DeSakar S C. Multiobjective heuristic search. Wiebaden: Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, 1999

[7] Mandow L, Perez de la Cruz J L. Multicriteria heuristic search. European Journal of Operational Research, 2003, 150(2): 253-280

[8] Mandow L, Perez de la Cruz J L. A new approach to multiobjective A* search//Proceedings of the 19th International Joint Conference on Artificial Intelligence. Denver, Co: Professional Book Center, 2005: 218-223

[9] Harikumar S, Kumar S. Iterative deepening multiobjective A*. Information Processing Letters, 1996, 58(1): 11-15

[10] Refanidis I, Vlahavas I. Multiobjective heuristic state-space planning. Artificial Intelligence, 2003, 145(1-2): 1-32

[11] Do M B, Kambhampati S. Sapa: A multi-objective metric temporal planner. Journal of Aritificial Intelligence Research, 2003, 20: 155-194

[12] Bryce D, Cushing W, Kambhampati S. Model-Lite planning: diverse multi-option plans and dynamic objective functions//Proceedings of the 3rd Workshop on Planning and Plan Execution for Real-World Systems (ICAPS'07). Menlo Park: AAAI, 2007: 93-100

[13] Wei Wei, Ouyang Dan-Tong, LU Shuai, YIN Ming-Hao. A multiobjective incremental heuristic search algorithm. Journal of Jilin University (Science Edition), 2009, 47(4): 752-758(in Chinese)
(魏唯，欧阳丹彤，吕帅，殷明浩. 一种多目标增量启发式搜索算法. 吉林大学学报(理学版), 2009, 47(4): 752-758)

[14] Bentley J L, Kung H T, Schkolnick M, Thomson C D. On the average number of maxima in a set of vectors and applications. Journal of the ACM, 1978, 25(4): 536-543

WEI Wei, born in 1984, Ph. D. candidate. Her main research interests include intelligence planning and automated reasoning.

OUYANG Dan-Tong, born in 1968, professor, Ph. D. supervisor. Her main research interests include model-based diagnosis and automated reasoning.

LU Shuai, born in 1981, Ph. D., lecturer. His main research interests include intelligence planning and automated reasoning.

FENG Yu-Xuan, born in 1980, Ph. D. candidate. His main research interests include intelligence planning and automated reasoning.

Background

The work focuses on the topic of solving multiobjective path planning problems under dynamic uncertain environment, which is supported by the National Natural Science Foundation of China under grant Nos. 60773097, 60873044, 60873148, 60973089, the Open Research Foundation of Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education of China under grant Nos. 93K-17-2009-K02, 93K-17-2009-K06 and Graduate Innovation Fund of Jilin University under grant No. 20111060.

Path planning is an important branch of many fields of artificial intelligence research. The task of optimal path planning is to find a best path in the state space between the start state problem and the goal state according to some criteria. The optimal path can be computed in a global planning process when the situation of the environment is static. However, sometimes it is impossible to know the complete information of the environment, especially the exact situation of dynamic obstacles, or the problems change their state space for the different situations, thus the moving path to the goal state must be recomputed by many local replanning processes when the environment situation is detected different from the original observation.

Only one criterion is considered in traditional path planning. But one single objective function can't describe several conflict criteria exactly. If the number of objective functions in a path planning problem is more than one, it is a multiobjective path planning problem. Multiobjective problems are different from single objective problems because of the conflict of the objective functions. A solution which is good for one objective function may be bad for another and this causes that there is usually a set of solutions composed of all the optimal paths. Due to the complexity of multiobjective situation, it is much more difficult to solve multiobjective path planning problems under dynamic uncertain environment.

In this paper, a method called DMPP is proposed which performs one global planning process and several local replanning processes. The experiment results show that the method adopting an incremental replanning can solve multiobjective path planning problems under dynamic uncertain environment efficiently by reusing the information of previous search and using heuristic information to lead the search process.