

基于 SOAR 架构的 A-STAR 寻路算法研究

丁 凡¹ 魏 畅¹ 王小鹏¹

DING Fan WEI Chang WANG Xiaopeng

摘 要

寻路算法是自动寻路过程中的重要因素, 诸如寻路规则、避障过程、三维空间坐标转换方式等多种因素都会对算法性能产生影响。文章通过对三维空间寻路算法的研究, 提出了一种基于 SOAR 智能架构的 A-STAR 算法, 通过将智能架构与寻路算法结合, 改善了自动寻路和避障的过程。通过仿真对比试验, 表明该算法对于寻路结果有较好的提升。

关键词

SOAR; A-STAR; 寻路算法; 智能架构; 三维坐标转换

doi: 10.3969/j.issn.1672-9528.2024.11.014

0 引言

随着飞机性能的不不断提升, 可探索的空域不断扩大, 更为复杂多变的空中环境也随之而来, 如何帮助飞行员找出一条具有安全、快速、油耗低等特点的线路就成了寻路问题的关键。多采用智能路径规划算法来解决寻路问题, 诸如基于采样的路径规划算法、基于节点的最优化算法、基于数学模型的算法、基于多重融合的算法, 以及基于生物启发的路径规划算法等。A-STAR 算法作为启发式路径规划算法的典型代表, 被广泛应用于各种寻路问题。

依靠单一的路径规划算法来解决寻路问题, 往往存在着规划线路非最优、进入死路和避障过程缓慢等问题。本文将 SOAR 智能架构与 A-STAR 算法相结合, 通过 SOAR 架构的长期记忆存储寻路过程与短期记忆存储条件的能力, 大大提升了 A-STAR 算法避障的速度和寻路条件的准确性, 一定程度上提升了 A-STAR 算法的性能。

1 SOAR 智能架构简介

SOAR 智能认知架构即 state-operator-and-result, 是一种由 John Laird、Paul Rosenbloom 和 Allen Newell 开发的一种主流的通用认知架构, 它通过符号主义对智能行为进行编码, 旨在描述和实现智能的基本功能组件。SOAR 模型通过将所有任务表述为状态空间搜索, 并使用分块作为唯一的学习机制, 其解决问题的基本思路是在工作内存中保留当前解决问题的环境, 之后通过选择不同的算子来改变当前工作状态, 以达到最终所需的目标状态。

SOAR 架构主要包含了长期记忆、短期记忆、I/O、决策和学习评估等部分, 具体架构如图 1 所示。

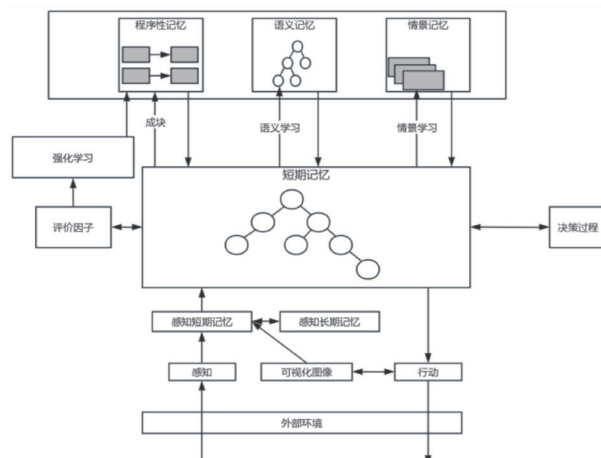


图 1 SOAR 智能架构图

外部环境通过 I/O 接口映射到短期记忆, 短期记忆记录了当前的工作状态, 同时从长期记忆获取知识和经验, 通过决策过程进行算子选择改变工作状态, 最后通过评价因子输出行为, 影响外部环境。

SOAR 架构提供的情景记忆和程序性记忆等长期记忆, 能够记录解决问题中的情景, 在短期记忆中通过决策过程和评价因子能够提升寻路条件的准确性。在寻路算法进入死路或避障过程时, 在保证最优条件的情况下快速复现之前寻路避障成功的情景, 大大降低算法在可行解空间的搜索时间, 提高了算法效率。基于 SOAR 架构的 A-STAR 算法具有避障快速、不易进入死路等优点, 适用于三维空间飞行路径规划的应用。

2 A-STAR 算法简介

A-STAR 算法是一种启发式路径规划算法, 通过不同的

1. 中国航空工业集团公司西安航空计算技术研究所
陕西西安 710065

启发函数来定义算法寻路过程中的代价消耗，不断循环迭代确定出一条最优路径。在三维空间寻路中，A-STAR 算法将空间划分为多个单位长度的立方体，每个立方体作为空间中的一个节点，所有节点共同构成了算法的搜索空间。

A-STAR 算法作为启发式算法，同时维护了开放列表和封闭列表，分别存储搜索空间中待访问的节点和已经访问过的节点，每个节点都有着自己的启发式函数值即带价值，所有节点的最优代价节点合集构成最终的最优路径。A-STAR 算法的启发式函数

$$f(n) = g(n) + h(n) \quad (1)$$

式中： $g(n)$ 表示起点到当前访问节点的实际代价值， $h(n)$ 表示起点到当前访问节点的估计代价值，通过启发函数可以控制函数的搜索行为。

(1) 当 $h(n)$ 接近 0 时，A-STAR 等同于迪杰斯特朗算法，能够保证一定有一条最优路径。

(2) 当 $h(n)$ 总是小于或等于从 n 节点到目标节点的步数时，A-STAR 一定能够找到最优路径，此时 $h(n)$ 越小，算法搜索时拓展路径越多，算法搜索速度越慢。

(3) 当估计代价值与实际代价值相等时，A-STAR 算法搜索路径时拓展的所有节点都在最优路径上，不会拓展任何无关节点，此时算法搜索速度最快，路径必为最优路径。

(4) 当 $h(n)$ 有时大于从节点 n 到目标节点的步数，那么 A-STAR 算法无法确保求得最优解，但会运行得更快。

(5) 当 $h(n)$ 非常接近于 $g(n)$ 时，A-STAR 算法等同于宽度优先搜索。

由此可见，A-STAR 算法的性能与启发式函数息息相关。

因此，本文提出一种改进的启发式函数：

$$h(n) = \partial(h'(n, x_1) + \text{dis}(x_1, x_2) + h'(x_2, \text{goal})) \quad (2)$$

式中： ∂ 是包含距离、油耗、威胁等条件信息的代价评估函数， h' 为新的估计函数， dis 代表距离代价， n 为其实际节点， x_1 为起始节点的临近拐点， goal 为目标节点， x_2 为目标节点的临近拐点。加入代价评估函数 ∂ 和估计函数 h' ，使得 A-STAR 算法的估计代价更为精确，使得估计代价 $h(n)$ 更加接近实际代价 $g(n)$ ，提高了算法的求解效率。

3 基于 SOAR 架构的算法构建

SOAR 架构作为一种主流的通用认知架构，能够用简洁的专属语法优化算法结构，利用独立的知识库进行编程来补充寻路规则，还能自动记录和处理僵局，与 A-STAR 算法结合能够提升算法的寻路效率与精度。

本文构建的基于 SOAR 架构的 A-STAR 算法可分为 SOAR 架构知识库、SOAR 接口程序和 A-STAR 算法，具体结构如图 2 所示。

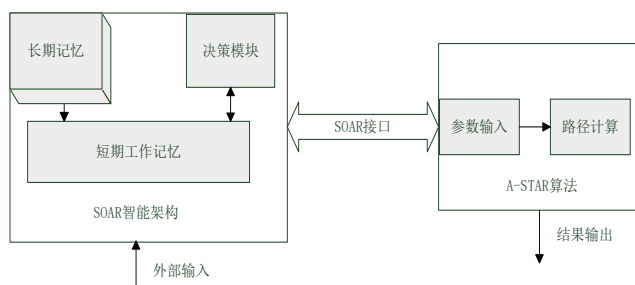


图 2 系统框架示意图

SOAR 智能架构部分通过 I/O 模块接收外部环境的参数输入和条件转变，在短期工作记忆内创建当前工作空间解决问题；调用长期记忆中记录的规则情景等知识处理问题；通过决策模块选择算子，将参数通过 SML 接口输出到 A-STAR 算法；算法接收参数后，通过启发式函数的不断迭代求解，最终求得一条最优路径。具体流程如图 3 所示。

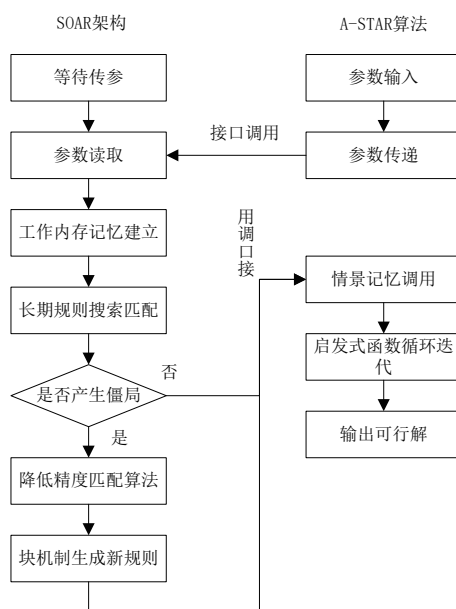


图 3 系统流程图

系统在 SOAR 智能架构中完成了 A-STAR 算法所需的各类代价、距离、拐点参数的计算，如果没有僵局产生，将上述参数通过 SOAR 的 SML 接口传给 A-STAR 算法进行后续计算。A-STAR 算法在搜索到死路或需要进行避障时能够通过接口调用 SOAR 的长期记忆，快速复现解决场景，大大提高了算法在搜索到死路和进行避障过程中的运行速度，提高算法效率。如果产生了僵局，SOAR 架构能够调用降低精度匹配算法来尝试以更少的编程规则继续当前的搜索场景，如果成功解决僵局，将解决过程生成块机制并存储在长期记忆中以供后续算法调用。本文提出的基于 SOAR 架构的 A-STAR 寻路算法的构建关键在于 SOAR 架构中长期记忆的调用、SOAR 架构编程对于算法规模的优化和独立性的提升以及三维空间坐标到节点的转换，具体如下。

3.1 调用长短期记忆

长期记忆能够在 SOAR 架构中长期保存,不随工作空间的释放而消失,分为语义记忆、情景记忆和程序性记忆。语义记忆中包含着有关外部世界的知识,能够帮助 SOAR 架构代理与外部环境进行交互。情景记忆中储存了工作记忆内的场景,用于记录过去解决问题的经验,通过对工作记忆图像进行学习,将其转化为代理经验并进行学习和使用。程序性记忆负责检索与控制和处理问题相关的知识,是对程序性知识进行编码的长期知识,存储的记忆记录着 SOAR 架构应该在什么时候解决问题以及如何解决问题。短期记忆中包含当前工作状态,通过决策过程和评价因子的配合能够更好地计算出路程代价。

本文构建的系统主要通过调用情景记忆和程序性记忆来提升算法搜索效率,配合短期记忆提升算法精度。将成功搜索到最优解的过程保存为图像化记忆,通过对图像化记忆的快速学习和搜索调用,提升了算法的搜索速度。

3.2 sp 规则与系统独立性的提升

通过将 A-STAR 算法使用 sp 规则进行编写,降低了算法规模,具体编写规则如下:

```
name
["documentation-string"]
[FLAG*]
LHS
->
RHS
```

sp 规则在左半部分描述条件,右半部描述行为来完成一个完整的规则表述,每个 sp 规则分为 propose 规则提议部分和 apply 规则应用部分。使用 sp 语言进行编写,使得 SOAR 架构能够直接识别 A-STAR 算法内容,进行之后的僵局处理和长期记忆调用。利用 sp 规则的特点,可以快速简单地实现 A-STAR 算法的编写,示例如下:

```
# 初始化 SOAR 工作内存所需规则,确定路径起始点
sp{propose*initialize-A-STAR
(state <s> ^superstate nil
  ^name)
-->
(<s> ^operator <o> +)
(<o> ^name initialize-A-STAR)
}
# 计算距离代价所需部分规则
sp{apply*h(n)
(state <s> ^name A-STAR
  ^io.input-link <i>
```

```
  ^io.output-link <o>)
  ^dis <d>)
(<i> ^numD <D>)
-->
(<s> distance (abs/((- <d> <D>) <D>)))
}
# 路径规划结束,释放工作内存所需部分规则
sp{apply*remove-A-STAR
(state <s> ^name A-STAR
  ^io.input-link <i>
  ^io.output-link <o>)
(<o> ^print <p>)
(<p> ^value <pr>)
-->
(<o> ^print.value <pr> -)
}
```

SOAR 架构通过 SML 结构与外部环境交联,采用 sp 规则编程,使得 A-STAR 算法与 SOAR 架构各自独立,互不影响,大大提升了系统整体的独立性。

3.3 三维空间坐标转换

三维空间寻路算法研究的一个关键点就是如何将三维空间转换为寻路过程中的单个节点,节点转换的精确程度直接影响寻路结果的准确与否。将经度、纬度、高度转变为更加直观的 XYZ 三系坐标,便于算法计算,转换时采用绕动态轴的转换方法,使得转换误差最小,保证了计算的精确度。

4 系统的验证与评价

本试验预设场景为三维空间中点到点的路径规划,构建地图边界为 200 km、高为 20 km,利用函数在地图中随机生成起始点、终点以及大小高度不一致的障碍物,寻路过程如图 4 所示。通过与单一 A-STAR 算法比较来验证系统的准确性和实时性,对比结果如图 5、图 6 所示。

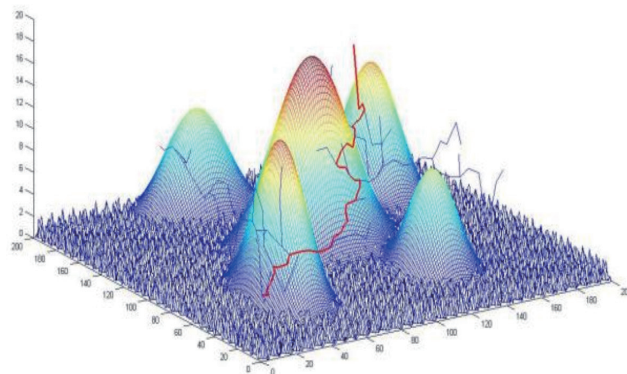


图 4 寻路示意图

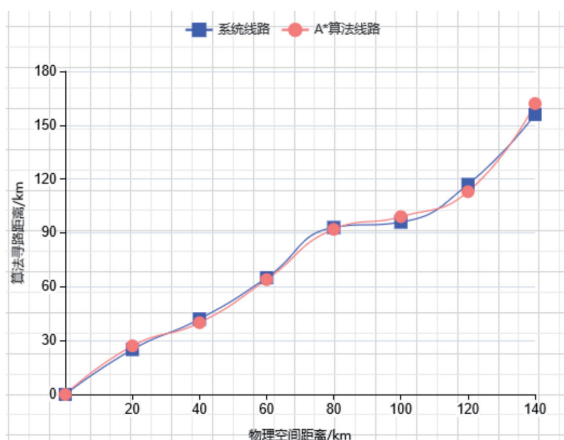


图 5 寻路结果对比

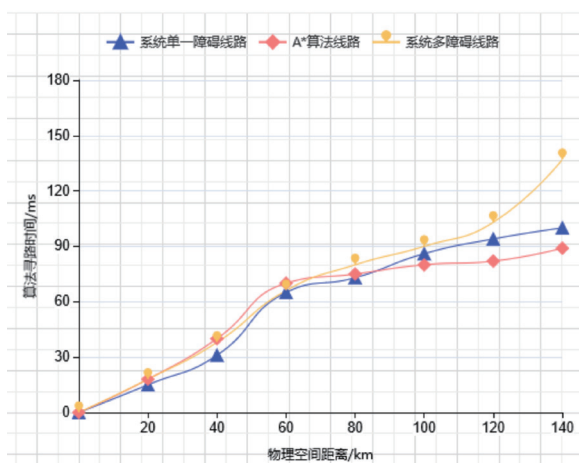


图 6 寻路所需时间

综上,基于 SOAR 架构的 A-STAR 算法能够在保持算法准确性的前提下,有效提升多障碍寻路的效率。

5 结语

本文使用 SOAR 智能架构构建了 A-STAR 寻路算法,利用 SOAR 架构的特性,大大降低了算法在避障过程中所消耗的时间,同时保持了算法寻路结果的准确性,对于 A-STAR 算法性能有一定提升。未来工作将针对三维空间转换进行进一步优化,提升寻路结果的精度。

参考文献:

- [1] 张浩杰,张玉东,梁荣敏,等.改进 A* 算法的机器人能耗最优路径规划方法[J].系统工程与电子技术,2023,45(2):513-520.
- [2] 张炳力,张成标,张羊阳,等.基于改进 A* 算法的铰接车辆混合路径规划方法:CN202310159677.9[P].2023-04-07.
- [3] LI Z, YING Y H, YANG M L, et al. Monitoring and path optimization of catalytic reformer in a refinery: Principal

component analysis and A* algorithm application[J].Expert systems with application, 2022,209(12):118358.1.

- [4] 丁凡,文鹏程,韩炜.基于 SOAR 模型的武器目标分配系统设计与研究[J].航空科学技术,2020,31(5):89-94.
- [5] ZUO G Y, PAN T T, ZHANG T L, et al. SOAR improved artificial neural network for multistep decision-making tasks[J]. Cognitive computation, 2021,13:612-625.
- [6] DANG C V, JUN M, SHIN Y B, et al. Application of modified Asimov's laws to the agent of home service robot using state, operator, and result (Soar) [J]. International journal of advanced robotic systems, 2018, 15(3):1-9.
- [7] YANG D, XU B, SHEN W H, et al. Passive Infrared (PIR)-Based indoor position tracking for smart homes using accessibility maps and A-Star algorithm[J].Sensors, 2018, 18(2):332.
- [8] WEI R X, XU Z F, WANG S L, et al. Self-optimization A-Star algorithm for UAV path planning based on Laguerre diagram[J]. International journal of advanced robotic systems, 2015,37(3):577-582.
- [9] KABIR R, WATANOB E Y, NARUSE K, et al. Effectiveness of robot motion block on A-Star Algorithm for robotic path planning[C]//Proceedings of the 21st International Conference on New Trends in Intelligent Software Methodologies, Tools and Techniques (SoMeT_22). Amsterdam: IOS Press, 2022:85-96.
- [10] ROVBO M A, SOROKOUMOV P S. Symbolic control system for a mobile robotic platform based on SOAR cognitive architecture[J]. Smart electromechanical systems, 2022, 13: 259-275.
- [11] PALOMBARINI J A, BARSCE J C, MARTINEZE E, et al. Generating rescheduling knowledge using reinforcement learning in a cognitive architecture[C/OL]. the Jornadas Argentinas de Informática (JAIIO).(2018-05-12)[2024-05-21].<https://doi.org/10.48550/arXiv.1805.04752>.

【作者简介】

丁凡(1995—),男,陕西西安人,硕士研究生,工程师,研究方向:嵌入式软件开发。

(收稿日期:2024-07-30)