

基于 Beam Search 算法的集装箱装载问题研究

杨会志

(电子科技大学中山学院 计算机学院, 广东 中山 528402)

摘要:针对具有优先装载约束的集装箱装载问题,对 Partial Beam Search 算法进行了改进。在搜索过程中去除相似中间状态,增加了搜索过程的多样性,提高了算法的搜索效率。实验结果证明了算法的有效性。

关键词:集装箱装载问题; Beam Search 算法; 优先装载约束; 搜索过程多样性

DOI: 10.11907/rjdk.151292

中图分类号: TP319

文献标识码: A

文章编号: 1672-7800(2015)007-0106-03

0 引言

在集装箱的实际装载过程中,除了考虑最大化空间利用率之外,还要考虑多种约束因素,如:货物的摆放方向、某些货物的隔离性、货物的承载能力、货物装载的稳定性、装载的优先级、集装箱的承重性等。目前,针对货物装载过程中具有优先装载约束的研究成果还比较少,主要有 Ren^[1]等人提出的基于树搜索的方法,在该方法中,采用 5 个评价函数选择简单块。Ning Wang^[2]等人提出的基于 Partial Beam Search 的方法,该方法改进了 Beam Search 算法,定义了更加高效的块选择函数和完整装载方案的评价函数,计算效果优于 Ren 等人提出的基于树搜索的方法。目前,满足 C_1 和 C_2 约束的集装箱装载问题的最好算法是 I. Araya^[3]等人提出的基于 Beam Search 的算法,该算法的最主要特点是运用 Beam Search 算法进行搜索时,通过去除搜索过程中产生的相似中间状态来提高搜索过程的多样性,进而提高搜索效率。

本文针对具有优先装载约束的集装箱装载问题,改进了 Partial Beam Search 算法,通过去除搜索过程中生成的相似中间状态来增加搜索过程的多样性,进而提高算法的搜索效率。

1 改进的具有优先装载约束的 Partial Beam Search 算法

根据 Wenbin Zhu^[4]等提出的集装箱装载算法分析框架,改进的具有优先装载约束的 Partial Beam Search 算法可以从以下 6 个方面进行描述。

1.1 空间表达方法

基于 cover representation 方法而不是 cut representation 方法进行装载空间表达,这是获得较高装载率的关键因素之一。

1.2 货物块种类

分别基于简单块和复合块,各用总体计算时间的一半来计算装载方案。在计算过程中不仅使用简单块,同时也使用复合块,这是获得较高装载率的关键因素之一。

及报送的实用性和可靠性。电网异常信息自动统计与报送统计平台的建设为调度员和管理部门及时掌握电网故障信息,快速准确作出调度决策提供了可靠的工具,对保障台风等自然灾害期间电网安全稳定运行有着重要意义。

参考文献:

- [1] 呼士召,陈钢,周巍,潮铸. “韦森特”台风期间广东电网调度应急处理及反思[J]. 广东电力, 2013, 26(6): 1-4.
- [2] 刘思捷,刘昌,李小燕,李嘉龙. 台风对广东电力负荷特性的影响

[J]. 云南电网技术, 2012, 40(1): 91-93.

- [3] PANDEY A K, BHUYAN, P MOHAPTRA D P. Component based exception handling framework for SOA based services[C]. 2014 International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE, 2014(2): 1-5.
- [4] 高勃. 基于 XML-RPC 技术的用户统一认证系统[J]. 计算机科学, 2012, 39(10): 178-180.
- [5] 陈钢,林少华,陈晓东,胡超. 广东电网调度运行电子发令系统的设计与实践[J]. 广东电力, 2013, 33(24): 88-90, 24.

(责任编辑:陈福时)

基金项目:中山市科学技术局项目(20123A315)

作者简介:杨会志(1969—),男,河北唐山人,博士,电子科技大学中山学院计算机学院教授,研究方向为数据库应用。

1.3 选择装载空间

选择最佳装载空间遵循以下标准:①选择具有最小 z 轴坐标的可用空间;②如果满足上述标准的空间有多个,则选择具有最小 Corner Distance 的可用装载空间,见图1;③如果满足上述标准的空间有多个,则选择其中体积最大的空间;④如果满足上述标准的空间有多个,假设 (x_1, y_1, z_1) 和 (x_2, y_2, z_2) 分别为可用装载空间的 corner 中离坐标原点最近和最远的点的坐标,按照 $(y_1, z_1, y_2, z_2, x_1, x_2)$ 的次序依次选择具有最小值的空间。

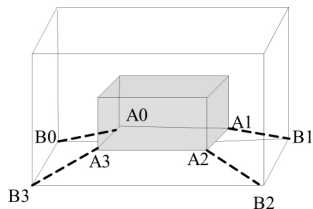


图1 可装载空间与集装箱之间的 Corner Distance

1.4 货物块选择策略

对状态 S ,在选择可用空间 s 后,可根据货物块的适应值最多选择 m 个货物块。设 $h(b)$ 和 $l(b)$ 分别为货物块 b 中优先装载的货物体积和非优先装载的货物体积。货物块的适应值函数 $f(s, b, S)$ 为:

$$f(s, b, S) = h(b) + (l(b) - waste) \cdot (1 - l(S)/max_{nh}) \quad (1)$$

其中 $waste$ 是货物块 b 装载到空间 s 产生的浪费空间近似值, $waste$ 项前面的负号表示浪费空间越大的货物块,其适应值越小。 $l(S)$ 是当前状态非优先装载货物的总体积, max_{nh} 是装载方案中非优先装载货物和浪费空间的体积之和。货物块适应值函数的第一项倾向于装载优先货物,在适应值函数的第二项中, $(1 - l(S)/max_{nh})$ 是一个随非优先装载货物被装载到集装箱而不断减少的权重系数。

1.5 货物块放置方式

当货物块 b 放入到可用装载空间后,放在 Anchor Corner,也就是具有最小 Corner Distance 的可装载空间的底角。

1.6 全局搜索算法

整体搜索过程采用基于 Double Search Effort 机制的 Partial Beam Search 算法。对一个给定的 $beamWidth$ 值,当 m 设置为一个很小的值时,常常会得到一个很差的解,而如果 m 取值太大,计算过程耗时又太长。因此,采用多次运行 Partial Beam Search 优化过程进行计算。计算开始时以很小的搜索代价 $r = m^2$ 进行计算,然后加倍 r 的值再次进行计算,连续进行此类过程直到计算时间结束,而 $beamWidth$ 作为算法的一个参数通过实验来确定它的最佳值。

Partial Beam Search 算法如图2所示。每一次的搜索以 $stateList$ 中包含一个代表空集装箱的状态开始。对每个 $stateList$ 中的状态 S ,如果它不是一个装载方案的最终

状态,算法将产生最多 $m = \lfloor \sqrt{r} \rfloor$ 个后继状态并加入到 $canList$ 中。在所有下一层的候选状态都生成之后,开始评价每一个候选状态。在第一轮搜索,初始化搜索代价 $r = 8$,这个值对应搜索宽度 $m = 2$ 。在一轮搜索完成后,开始把 r 值加倍,得到 m 值后开始下一轮搜索。在给定的计算时间内连续进行多轮搜索,然后返回,找到最好的装载方案。

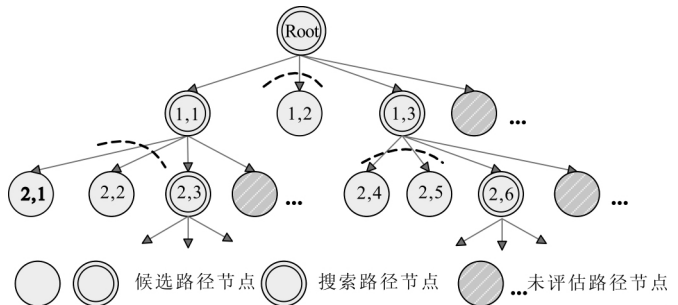


图2 Partial Beam Search 算法

集装箱装载的 Partial Beam Search 算法的伪代码如下:

```
MultiRoundPartialBeamSearch (BlockType, beamWidth,
timeLimit)
    blockList = Generate blocks according to BlockType
    bestSol = NULL; r = 8
    while accumulated CPU time not exceeding timeLimit
         $m = \lfloor \sqrt{r} \rfloor$ 
        stateList = {initialstate}
        while stateList not empty
            candList = NULL
            for each state S in stateList
                remove S from stateList
                if S is a terminal state
                    update bestSol if S represents better solution
                if S is not a terminal state
                    Generate up to m successors of S and append them to
                    candList
            evaluate each state in candList
            stateList = top beamWidth states in candList
             $r = r \times 2$ 
    return bestSol
```

评价在状态 S 下生成的 m 个后继节点,采用一种深度受限的树搜索算法,如图3所示。 $root$ 节点对应要装载的一个货物块,对每个 $root$ 节点的后继节点状态的评价,采用连续装载适应值最好的货物块策略,直到完成所有装载过程。在得到所有的完整装载方案后,装载方案的评价函数由装载方案中优先装载货物的体积和装载方案总体积组成数据对,然后对所有装载方案所对应的数据对值进行降序排序,排序越靠前,函数值就越高。为了避免搜索过程收敛到相似的状态,提高搜索过程的多样性,在搜索过程生成新状态的集合 S 中去除相似的状态。规定如果搜索过程中的两个中间状态通过贪婪过程得到的装

载方案包括完全相同的货物,则认为状态 S_1 和 S_2 是相似的,如果出现这种情况,那么在新的状态集合 S 中只保留那个货物体积最小的中间状态。

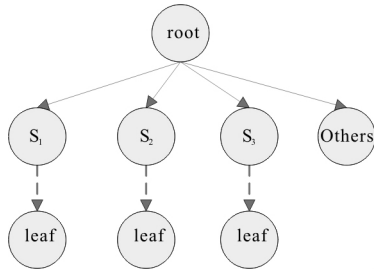


图 3 时对后继节点进行评价

2 计算结果

表 1 是在 BR 数据集合上的改进算法与原有算法计算结果的对比情况,Partial Beam Search 算法用 BSSOLVER 标识,本文算法用 BSSOLVER' 标识。每个数据组包含 100 个实例,装载率的值是 100 个实例装载率的平均值。在 Intel 酷睿 2 双核 T7200,2G 内存计算机上,每个实例运行和 BSSOLVER 算法相同的时间 400s。从表 1 可以看出,BSSOLVER' 算法在计算机性能较低的情况下,计算结果优于 BSSOLVER 算法,平均装载率提高了 0.27%。

3 结语

本文针对具有优先装载约束的集装箱装载问题,在集

装箱装载的 Partial Beam Search 算法基础上进行了改进,通过去除搜索过程中生成的相似中间状态来增加搜索过程的多样性,有效提高了算法的搜索效率。今后的研究方向是提高搜索过程的多样性。

表 1 在 BR 数据集合上的计算结果对比

测试集合	优先装载	非优先装载	BSSOLVER (%)	BSSOLVER' (%)
BR1_1	1	2,3	94.28	94.55
BR1_2	3	1,2	94.15	94.38
BR4_1	1,2,...,5	6,7,...,10	94.35	94.60
BR4_2	6,7,...,10	1,2,...,5	94.22	94.55
BR7_1	1,2,...,10	11,12,...,20	93.12	93.35
BR7_2	11,12,...,20	1,2,...,10	93.10	93.21
平均值			93.84	94.11

参考文献:

- [1] REN J ,TIAN Y ,SAWARAGI T. A tree search method for the container loading problem with shipment priority[J]. European Journal of Operational Research,2011:526-535.
- [2] WANG N,LIM A,ZHU W. A multi-round partial beam search approach for the single container loading problem with shipment priority[J]. International Journal of Production Economics,2013, 145(2): 531-540.
- [3] I ARAYA,M C RIFF. A beam search approach to the container loading problem[J]. Computers & Operations Research, 2014 (43):100-107.
- [4] ZHU W,OONW,LIM A,et al. The six elements to block-build approaches for the single container loading problem[J]. Applied Intelligence,2012,37(3):1-15.

(责任编辑:杜能钢)

A Beam Search Approach for the Single Container Loading Problem

Abstract: To solve the single container loading problem with shipping priority a new algorithm is developed based on a beam search approach. The beam search based strategy maintains a set of different promising paths and not only one as most of the building based approaches. The competition among different states allows our algorithm to quickly discard worse states, opening the search to more promising states. Also a simple mechanism for removing similar states maintains a minimum required level of diversity among the states. Results show that the new approach outperforms all the others for each set of instances.

Key Words: Container Loading Problem; Beam Search; Shipment Priority; Diversity