

可用性约束资源预留与分配算法

李 波 石冰心 沈 斌

(华中科技大学电子与信息工程系 武汉 430074)

摘 要 给定一系列作业和只能在有限的时间段可用的资源,如何预留和分配资源以实现作业的最大完成时间最小化的问题是 NP 难的。本文将其归结为一种新型的尺寸可变装箱问题并给出了作业信息和资源信息完全已知条件下的六种离线算法,理论分析表明所给算法的渐进最坏比为 2,在作业相互独立的条件下推广的降序最佳适合(Best Fit Decreasing)算法的平均性能最优,在作业有先后依赖关系的条件下推广的最佳适合(Best Fit)算法的平均性能最优。

关键词 资源预留与分配,装箱问题,离线算法,可用性约束

Algorithms for Resource Reservation and Allocation with Limited Availability

LI Bo SHI Bing-Xin SHEN Bin

(Department of Electronics and Information Engineering, Huazhong University of Science & Technology, Wuhan 430074)

Abstract Given a List of jobs and a resource with limited available intervals, it is a NP-hard problem to decide how to reserve and allocate available intervals of the resource to the jobs with the goal to minimize the maximum completion time of jobs. The paper attempts to define a mathematical model for the problem in terms of a Variant of the Variable-Sized Bin Packing problem and six algorithms in the classic bin packing problem are adapted for the offline version of the VVSBP problem. Analytical results show that the competitive ratios of the adapted algorithms are of 2. Experimental results for average cases show the adapted First Fit(Decreasing) algorithm outperforms any others when jobs are precedence-constrained(independent).

Keywords Reservation and allocation, Bin packing, Offline algorithms, Availability constraints

1 引言

资源预留和分配是在资源难以满足所有应用的需求的条件下保证部分作业的服务质量的重要手段,是资源管理策略的重要组成部分。在以往的研究中,一般假定预留的资源能够连续地处理完一个应用所包括的所有作业。但是,由于资源维护或者需要周期性地执行高优先级作业,资源并不能保证可以在一段很长的连续的时间内对作业进行处理,即资源不是持续可用的。如果作业的处理过程可以在中断后从中断点继续执行,资源的可用性的不连续性对资源分配影响不大,只需要在满足应用的服务质量要求的前提下分配作业所需的资源总量即可,如果作业的处理过程在中断之后必须重新开始或者需要重复处理在中断前已经处理过的部分任务,资源可用性的连续与否将对作业的服务质量和所占用的资源总量产生影响,后者将直接影响到资源的服务效率和系统容量。本文将研究在资源可用性不连续条件下如何为那些处理过程中断后就必须重新开始的作业预留和分配资源从而实现给定作业系列的最大完成时间最小化的问题。在网格和移动计算之类的动态计算环境中,由于资源的动态性和自治性,资源的可用性约束对于作业的分配和执行过程的影响将比在小规模专用系统中更明显,基于可用性约束的资源预留和作业分配技术将有广泛的应用前景^[1~3]。此外,该研究也可用于可用性约束的柔性制造和作业调度系统^[4]。

在上述问题中,假定资源的第 i 个可用时间段及其长度分别对应第 i 个箱子 b_i 及其长度 $s(b_i)$, $i \geq 1$,而作业系列对应装箱问题中的物品系列 $L = (a_1, a_2, \dots, a_n)$,各作业所需处理时间 t_j , $1 \leq j \leq n$ 对应物品长度 $s(a_j)$, $1 \leq j \leq n$,则此问题可以等价待装箱物品系列在装箱前已准备好,物品不能分割,各种尺寸的箱子顺序到达,要求将所有物品装入到箱子中并实现从第一个箱子到最后一个被使用的箱子为止的所有的箱子

尺寸总和最小化的装箱问题。此装箱问题是尺寸可变装箱 VSBP^[5](Variable-Sized Bin Packing)问题的变体,同样也是一个 NP-Hard 问题。为了便于描述,下面将其统称为变形的尺寸可变装箱 VVSBP(Variant of VSBP)问题。它具有装箱问题的基本特征,主要区别在于:(1)在 VVSBP 问题中假定物品是装箱前准备好的,而箱子是顺序出现的,因此是否在装箱过程中对物品按照尺寸大小排序并不决定算法是否为在线算法。经典的装箱问题和尺寸可变装箱问题中箱子可看作是装箱前准备好的,而物品是顺序出现的,只有离线算法才能在装箱前对物品按照尺寸大小排序。(2)尽管优化目标都是实现箱子长度总和最小化,但是目标函数的计算方法不同。在 VVSBP 问题中,从第一个箱子到最后一个被使用的箱子为止的所有的箱子,不论其是否装入了物品,在目标函数中其长度都被计算在内,在装箱过程中选择大小相同但是出现顺序不同的箱子可能对目标函数产生影响。在经典的装箱问题和尺寸可变装箱问题中,只有被实际装入了物品的箱子的长度才被计算在目标函数中,将一个箱子中的所有物品换到另一个尺寸相同的箱子对目标函数无影响。

如果所有可用时间段的长度不小于最大作业处理时间并且在可用时间段到达前不能预测,在每个可用时间段到达时必须决定将哪些作业分配到此时间段,这对应了在线的 VVSBP 装箱问题,可用文[6,7]提出了在线算法进行求解。如果在资源预留和分配之前所有作业信息和可用时间段信息已知,这对应了离线的 VVSBP 装箱问题。根据作业之间是否有先后约束,本文分别给出了三种算法并对这些算法的最坏性能进行了理论分析,结果表明所有算法的渐进最坏比(也称为竞争比)为 2。为了区分这些算法的平均性能,本文还将算法在物品和箱子尺寸为随机连续均匀分布条件下的问题实例进行了模拟试验,结果表明在作业相互独立的条件下降序最佳适合(Best Fit Decreasing)算法的平均性能最优,在作业有先

后依赖关系的条件下最佳适合(Best Fit)算法的平均性能最优。

本文在第2节给出了六种离线算法,然后在第3、第4节分别对算法的最坏性能和平均性能分别进行了研究,最后是总结。

2 离线算法

在经典的一维装箱问题 BP(Bin Packing)中,可以假定在装箱前箱子的信息已知而物品顺序到达,一旦装箱前所有物品信息已知,我们就可以通过对物品按照尺寸不减排序然后对排序后的物品使用各种在线算法,就可以得到离线算法^[8,9]。在标准的 VSBP 问题中,到目前为止仅有很少的关于离线算法的研究成果,在已有的离线算法中需要采取重新装箱(repacking)或者转移(shifting)的方法将物品从较大的箱子换到较小的箱子来减少占用的箱子的尺寸^[9,10]。由于前面所述的 VVSBP 问题与这两种装箱问题的差别,我们不能将这两种问题中已有的离线算法直接用于 VVSBP 的离线装箱问题。

除了所有物品信息之外,VVSBP 问题的离线算法需要在装箱前已知所有箱子的信息,因此可以将 BP 问题中能同时保持无限个箱子为打开状态的 Any Fit 算法引入到 VVSBP 问题中。参照经典的一维装箱问题中各算法的定义,下面给出各算法在 VVSBP 中的对应算法并且用算法的原有名称表示其在 VVSBP 中的对应算法。

最佳适合(Best Fit)算法:对物品系列 L ,从 a_1 开始按照物品顺序进行装箱,当前物品 a_j 的长度为 $s(a_j)$, $1 \leq j \leq n$ 且当前打开的序号最大的箱子为 b_k ,如果 $s(a_j)$ 不大于 $b_1 \sim b_k$ 的剩余空间,则将其放入到序号最小且容纳 a_j 后剩余空间最小的箱子中。如果这样的箱子不存在,则打开 b_{k+1} 并将当前物品装入其中。

最坏适合(Worst Fit)算法:与 BF 算法不同的是,当前物品首先将被试图放入剩余空间最大且序号最小的打开的箱子中。

几乎最坏适合(Almost Worst Fit)算法:与 WF 算法不同的是,当前物品首先将被试图放入剩余空间次最大且序号最小的打开的箱子中。

上述算法用于作业需要按照在系列中的给定顺序进行处理的情形。如果作业是相互独立的,通过在装箱之前将物品按照尺寸不增的顺序排序,对排序后的物品系列再使用上述 BF、WF 和 AWF 算法,则分别得到降序最佳适合(Best Fit Decreasing)算法、降序最坏适合(Worst Fit Decreasing)算法和降序几乎最坏适合(Almost Worst Fit Decreasing)算法。

3 最坏情形分析

在装箱算法研究中,常从特定算法用于给定的一类问题时的结果与最优结果的最大偏离程度和平均偏离程度两方面来对算法的性能进行评价,前者称为最坏情形性能,量化指标一般采用渐进最坏比,后者称为平均情形性能,量化指标一般为渐进期望比或者平均浪费空间^[8,9,11]。本文的装箱模型与其他的装箱模型不同的是,装箱过程同时与物品系列和箱子系列有关,在此将装箱问题中原有的渐进最坏比的定义修改如下:对于物品系列 L 、箱子系列 B 和近似算法 H ,分别用 $H(L, B)$ 和 $OPT(L, B)$ 表示算法 H 和最优装箱所用的总的箱子尺寸,定义算法 H 的渐进最坏比为 $R_H^\infty = \limsup_{k \rightarrow \infty, L, B} \{H(L, B) / OPT(L, B) | OPT(L, B) = k\}$ 。 R_H^∞ 越小,则认为算法 H 的性能越好。令算法 $H \in \{BF, WF, AWF, BFD, WFD, AWFD\}$,

在 VVSBP 问题中其最坏情形性能如下:

定理 1 对于所有实例, $H(L, B) \leq 2OPT(L, B) + 1$ 。

证明:假定算法 H 所得装箱方案中最后装入物品的箱子的序号为 k ,对于 $1 \leq i \leq k-1$,有 $s(b_i) < |b_i| + |b_{i+1}|$,其中 $|b_i|$, $1 \leq i \leq k$ 为箱子 b_i 中已经装入的物品长度之和,则有 $H(L, B) = \sum_{i=1}^k s(b_i) < 2 \sum_{i=1}^k |b_i| - |b_k| + s(b_k)$,由于 $\sum_{i=1}^k |b_i| = \sum_{j=1}^n s(a_j)$ 且 $\sum_{j=1}^n s(a_j) \leq OPT(L, B)$,所以有 $H(L, B) \leq 2OPT(L, B) + 1$ 。

定理 2 对于所有实例, $R_H^\infty = 2$ 。

证明:借鉴文[6]证明在线算法的渐进最坏比的过程中所用的问题实例,令 $\delta < 2^{-(n+2)}$ 为一个充分小的正数,定义 $\epsilon_0 = 0$, $\epsilon_j = \frac{(j+1)!}{(j+1)! + 1}$, $j = 1, 2, \dots, n$, 有 $0 = \epsilon_0 < \epsilon_1 < \dots < \epsilon_n < \delta$ 且 $\epsilon_n + \epsilon_j < 2\epsilon_{j+1}$, $j = 1, 2, \dots, n-2$ 。假定物品系列 L 中的物品按照尺寸大小分为 $n+2$ 组,每组物品尺寸相同,第一组 I^1 包括一个尺寸为 $0.5 + \delta$ 的物品,第二组 I^2 包括两个尺寸为 0.5 的物品,第 j 组 I^j , $3 \leq j \leq n+2$ 包括 2^{j-2} 个尺寸为 $0.5(1 - \epsilon_{j-2})$ 的物品。同时假定所有物品在 L 中按尺寸不增排序。同理,假定箱子系列 B 包括 $n+3$ 种尺寸的箱子,第一种 B^1 包括一个尺寸为 1 的箱子,第二种 B^2 包括一个尺寸为 $0.5 + \delta$ 的箱子,第 i 种 B^i , $3 \leq i \leq n+2$ 包括 2^{i-3} 个尺寸为 $(1 - \epsilon_{i-2})$ 的箱子,最后一种 B^{n+3} 包括 2^n 个尺寸为 $1 - \delta$ 的箱子。同时假定所有箱子在 B 中按照尺寸不增排序。在优化的装箱过程中,将 a_1 装入 b_2 , a_2 和 a_3 装入 b_1 ,对于 $3 \leq j \leq n+2$ 将两个 I^j 组的物品装入到一个 B^j 种的箱子中,所用的总的箱子尺寸 $OPT(L, B) = 1 + (0.5 + \delta) + \sum_{i=2}^{n+1} 2^i(1 - \epsilon_{i+1}) < 2^n + 1$ 。在算法 H 的装箱过程中,将 a_1 装入 b_1 , a_2 装入 b_2 , a_3 装入 b_3 ,对于 $3 \leq j \leq n+2$,将一个 I^j 组的物品装入到一个 B^{j+1} 种的箱子中,最后类型为 B^{n+3} 的所有箱子都将被装入物品, $H(L, B) = OPT(L, B) + 2^n(1 - \delta) > 2^{n+1} - 1$ 。至此可知 $H(L, B) / OPT(L, B) > (2^{n+1} - 1) / (2^n + 1)$ 。当 n 足够大,可得 $R_H^\infty \geq 2$ 。同时考虑定理 1 的结论,必然有 $R_H^\infty = 2$ 。

表 1 列出了各算法在 BP 问题^[8]和 VVSBP 问题中的复杂度和渐进最坏比,可以看出这些算法在 VVSBP 问题中的渐进最坏比都相等,这与这些算法在 BP 问题中最坏性能差别很大。

表 1 算法的复杂度及其渐进最坏比

算法	复杂度	渐进最坏比	
		BP	VVSBP
BF	$O(n \log n)$	1.7	2
BFD	$O(n \log n)$	11/9	2
WF	$O(n \log n)$	2	2
WFD	$O(n \log n)$	5/4	2
AWF	$O(n \log n)$	1.7	2
AWFD	$O(n \log n)$	无结论	2

4 一般情形性能

相对于最坏情形性能分析,对算法的平均性能进行理论分析要困难得多。除了理论分析之外,更多的是采用试验模拟的方法对算法的平均性能进行统计分析。在经典的在线装箱问题中,可假定所有箱子尺寸都归一化为 1,并且已在装箱前准备好,各种归一化尺寸小于 1 的物品顺序到达。通过假定物品归一化尺寸的概率分布(一般为指数分布、连续均匀分布或者离散均匀分布),然后就可以用概率分析的方法对各种算法的渐进期望比、平均浪费空间等性能参数进行研究^[8,9,11]。在本文的装箱模型中,由于需要同时定义箱子尺寸和物品尺寸

的概率分布,所以对各算法的概率分析将更困难。本文通过对不同算法在物品尺寸和箱子可用长度都是连续均匀分布的条件下的平均性能进行数值模拟研究,这种研究方法曾在文[12]对经典装箱问题中有限空间的在线算法的平均性能研究中被采用。由于难以对模拟过程中的各个实例求出最优化方案,因此在模拟过程中不采用最优解来对不同算法的性能进行比较。定义算法 H 的性能指标如下:(1)绝对平均性能 $E(BI_H)$,其中 $BI_H = \sum_{i=1}^k s(b_i) / \sum_{i=1}^n s(a_i)$, k 为算法 H 在一次试验中所使用的箱子的序号的最大值。 $E(BI_H)$ 为 m 次试验中算法 H 的 BI_H 值的平均值。(2)相对平均性能 $OptRate$,在 m 次试验中,算法 H 的 BI_H 值为所有参与试验的算法中最小的次数与 m 次试验中所有算法都能得到可行解的次数的比值。

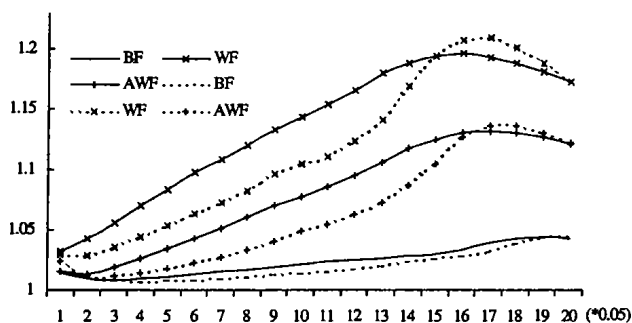
对上述各种算法的绝对平均性能和相对平均性能的数值模拟过程如下:

(1)生成 n 个在 $[0, u]$, $0 < u \leq 1$ 均匀分布的随机数构成物品长度系列 $L_{n,u}$;

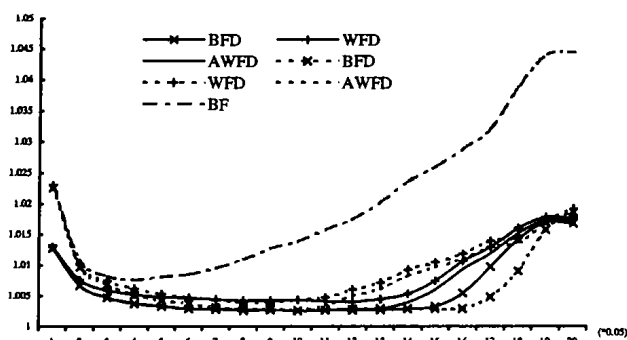
(2)生成 n 个在 $[u, 1]$ 均匀分布的随机数构成箱子长度系列 $B_{n,u}$ 。

(3)将各算法用于上述生成的物品系列和箱子系列的装箱过程,计算其 BI_H 。

(4)重复(1)~(3),共 m 次,统计不同算法在 m 次试验中的 $E(BI_H)$ 和 $OptRate$ 。



(a) 各算法随 u 变化的绝对性能



(b) 各算法随 u 变化的绝对性能

图 1

从渐进性能的定义可知,系列长度越长,试验次数越多,所得模拟结果便越接近理论值。从各算法在经典的一维装箱问题的性能可以预测,在 VVSBP 问题中 BF 算法应该优于 AWF 算法,而 AWF 算法优于 WF 算法,同理 BFD 算法优于 AWFd 算法,而 AWFd 算法优于 WFD 算法。本文实验结果证实了这种预测。图 1(a)给出了 BF、WF 和 AWF 算法随 u 变化的绝对性能曲线,模拟过程中系列长度 n 为 1000, u 取 0.05 到 1.0,步长为 0.05,每个数据点为 100 个实例的平均值。另外,实验结果表明 BFD、WFD 和 AWFd 算法明显优于 BF、WF 和 AWF 算法。为了清晰地表示它们的性能,将其在

图 1(b)中给出。为了和各算法在经典一维装箱问题中的性能进行比较,令步骤(2)中 $u=1$,本文得到了用于各个算法在经典的一维装箱问题中的平均性能,相应结果图中用虚线表示。

从图中可以看出,所有算法在经典装箱问题和 VVSBP 问题中的绝对性能都不是随 u 单调变化的,而且在经典装箱问题中的绝对性能并不始终比在 VVSBP 问题中的绝对性能更好。与绝对性能曲线的结论相补充,图 3 给出了各算法的相对性能曲线,实线表示 VVSBP 中的结果,虚线表示当算法用于经典一维装箱问题中的实例时的相对性能曲线。

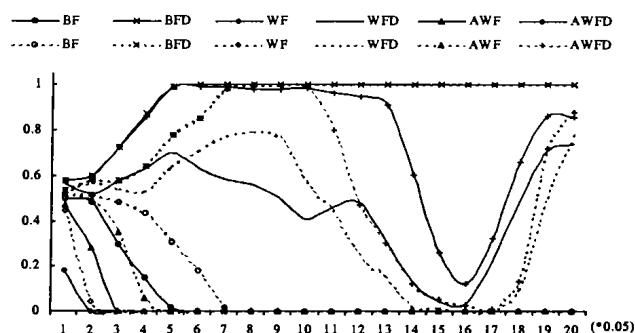


图 2 各算法随 u 变化的相对性能

综合以上所有结论可以看出各算法的相对性能的差别,其中 BFD 是性能最优的算法,AWFd 和 WFD 次之。如果不对物品在装箱前排序,则 BF 算法性能最优,AWF 次之,WF 最差。

结论 本文用装箱问题模型和算法研究了给定一系列作业和只能在有限的时间段可用的资源,如何预留和分配资源的可用时间段以实现作业的最大完成时间最小化的问题。通过将经典的一维装箱问题中的算法推广到此问题中,本文得到了六种离线算法,理论分析表明所给的算法的渐进最坏比为 2,在作业相互独立的条件下降序最佳适合(Best Fit Decreasing)算法的平均性能最优,在作业有先后依赖关系条件下最佳适合(Best Fit)算法的平均性能最优。

参考文献

- 1 Foster I, Kesselman C, Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations [J]. Int'l J. High-performance Computing Application, 2001, 15(3): 200~222
- 2 Foster I, Roy A, Sander V. A Quality of Service Architecture that Combines Resource Reservation and Application Adaptation [A]. In: Proc. of the 8th Intl. Workshop on Quality of Service (IWQOS) [C]. Pittsburgh, PA. 2000. 181~188
- 3 Hovestadt M, Kao O, Keller A, et al. Scheduling in HPC Resource Management Systems: Queuing vs. Planning [A]. In: Proc. of the 9th Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP) [C]. Seattle, USA. 2002
- 4 Schmidt G. Scheduling with limited machine availability [J]. European Journal of Operational Research, 2000, 121(1): 1~15
- 5 Friesen D K, Langston M A. Variable sized bin packing [J]. SIAM journal on computing, 1986, 15: 222~230
- 6 Zhang G. A new version of on-line variable-sized bin packing [J]. Discrete Applied Mathematics, 1997, 72(3): 193~197
- 7 Xing W, Lam K. Capacitated single machine scheduling and its on-line heuristics [J]. IIE Transactions, 2002, 34(11): 991~998
- 8 Coffman E G, Jr, Garey M R, Johnson D S. Approximation Algorithms for Bin Packing: A Survey [A]. In: Hochbaum D. ed. Approximation Algorithms for NP-Hard Problems [M]. Boston, MA: PWS Publishing, 1996. 46~93
- 9 Coffman E G, Jr, Galambos G, Martello S, et al. Bin Packing Approximation Algorithms: Combinatorial Analysis [A]. In: Du D.-Z., Pardalos P. M. eds. Handbook of Combinatorial Optimization [M]. Kluwer Academic Publishers, 1998
- 10 Kang J, Park S. Algorithms for the variable sized bin packing problem [J]. European Journal of Operational Research, 2003, 147(2): 365~372
- 11 Coffman E G, Jr, Courcoubetis C, Garey M R, et al. Perfect Packing Theorems and the Average Case Behavior of Optimal and On-line Bin Packing [J]. SIAM Review, 2002, 44(1): 95~108
- 12 Csirik J, Johnson D S. Bounded Space On-Line Bin Packing: Best is Better than First [J]. Algorithmica, 2001, 31(2): 115~138