

一种启发式混合遗传算法及其在 车间作业排序问题中的应用

周 泓 冯允成

(北京航空航天大学管理学院, 北京, 100083)

A KIND OF HYBRID HEURISTIC GENETIC ALGORITHM WITH APPLICATIONS TO JOB SHOP SCHEDULING

Zhou Hong, Feng Yuncheng

(Management School, Beijing University of Aeronautics and Astronautics, Beijing, 100083)

摘 要 针对一般车间作业排序问题($n/m/G/C_{\max}$) 建立了一种启发式混合遗传算法, 将调度规则 SPT 与 MWKR 引入遗传搜索过程, 并应用邻域搜索法作局部辅助搜索, 以提高解的性能。通过与目前一些常用算法的比较表明, 该算法具有良好的求解性能与运算效率。

关键词 排序 生产规划 启发式算法 遗传算法

中图分类号 O223, C931.1

Abstract A kind of hybrid heuristic genetic algorithm is introduced for the job shop problem $n/m/G/C_{\max}$, in which the scheduling rules of SPT and MWKR are applied, and the neighbourhood search technique (NST) is adopted as an auxiliary procedure to improve the solution performance. The new algorithm is proved to be both effective and efficient by the comparison with some methods in common use at present.

Key words sequencing, production planning, heuristic approach, genetic algorithm

车间作业排序问题所要解决的是确定每台机器上各工件的投入顺序与投入时间, 该顺序必须与技术约束相容(即符合各工件加工工序流程的要求), 并使某一性能指标达到最优。这一问题是个典型的 NP 难题^[1], 即使规模较小时也很难求出其精确的最优解, 近年发展起来的遗传算法^[2]对该类问题的求解具有较为显著的优势。

遗传算法的一个重要特征是算法不依赖于问题的具体形式, 即具有很高的稳健性。但稳健性与算法效率之间往往存在着一种折衷关系, 较高的稳健性实际上导致了较低的运算效率, 而在很多情况下, 运算效率对一个算法来说更为重要。研究表明, 针对所求解问题的具体特点来设计遗传算法, 将对提高运算效率起重要作用^[3]。本文将作业排序中的“最短加工时间优先规则(SPT)”与“最大剩余工作量优先规则(MWKR)”引入遗传算法, 并利用邻域搜索法(NST)在遗传算法的基础上做局部辅助搜索, 以提高求解质量。通过大量的算例分析, 以及与当前求解作业排序问题的常用算法相比较, 可以看出, 本文所提出的启发式混合遗传算法在求解性能和效率两方面均具有显著的优势。

1 算法模型

本文研究的车间作业排序问题如下:某车间配备有 m 台机器, 将要加工 n 种工件, 每种工件的加工过程可由一系列工序组成, 每道工序由相应的机器完成, 每台机器同时只能加工 1 个工件, 而且机器在加工过程中不能被其它工件优先占用。排序问题所要达到的目标是使全部工件的完工时间(C_{max})为最短。根据文献[4]对排序问题的分类, 本文所研究的问题可简记为 $n/ m/ G/ C_{max}$ 。该问题的解实际上是每台机器上各工件加工顺序的排列, 即

$$\begin{array}{l}
 M_1 : O_{11} \quad O_{12} \quad \cdots \quad O_{1n} \\
 M_2 : O_{21} \quad O_{22} \quad \cdots \quad O_{2n} \\
 \text{-----} \\
 M_m : O_{m1} \quad O_{m2} \quad \cdots \quad O_{mn}
 \end{array}$$

其中: O_{ij} 为机器 M_i 上的第 j 项加工操作, $O_{ij} \in \{ J_1, J_2, \dots, J_n \}$; M_i 表示机器 i ; J_j 表示工件 j 。

以上问题的解空间规模(所有可行解与不可行解)为 $(n!)^m$ 。显然, 即使 m 与 n 都不大(例如 10), 我们也无法遍历整个解空间去搜索最优解。GA 对这类问题有较强的求解能力, 但遗憾的是求解效率不够高, 若想搜索到较好的解, 必须要有很长的进化时间, 这一点对于规模较大的问题尤为突出。因此, 有必要引入启发式的思想和方法, 用以提高搜索效率。

本文通过大量的数值实验发现, 每台机器上的第 1 道工序 $O_{i1}(i= 1, 2, \dots, m)$ 在决定排序性能指标方面起着重要作用, 若对第 1 道工序进行搜索寻优, 而对后续工序则采用某些启发式调度规则(如 SPT 或 MWKR 等)来确定, 其结果也是相当令人满意的。文献[5, 6]对各种调度规则及其适用条件进行了广泛深入的研究, 发现 MWKR 规则对于优化目标为 C_{max} 类型的问题具有最好的性能, SPT 则次之。在将 MWKR 规则应用于 GA 时, 算法将选择一个剩余加工时间(未完成工序的总时间)最长的工件优先加工; 而应用 SPT 规则时, 在当前机器上加工时间最短的工件将被优先加工。通过引入以上的启发式调度规则, 所要搜索的解空间规模减至 n^m 。此外, 这种混合方法还能在很大程度上保证解的可行性, 尽量减少“死锁排序”(与技术约束不相容因而无法实现的排序)。而一般的遗传算法在执行过程中常常产生大量的不可行解, 往往需花费很多时间去处理它们。因此, 本文的混合遗传算法其运行效率比一般遗传算法要高得多。然而, 调度规则在提高算法效率的同时, 也不可避免地带来了一些弊病, 即搜索空间的收缩可能会漏掉更好的解。为弥补这一缺陷, 本文引入了邻域搜索法(NST)作为遗传算法的辅助搜索工具。总的说来, 本文的核心思想就是, 先利用启发式调度规则将问题的解空间缩至某个包含高性能解的子空间上, 再进行遗传搜索, 然后应用邻域搜索法来寻找可能落在子空间外的更好的解。

本文算法的核心部分是遗传搜索。将遗传算法用于作业排序问题时需做某些必要的修正, 现简述如下。

(1)染色体编码方式 传统的遗传算法中, 最基本也最常用的染色体编码方式是利用二进制数 0 与 1 来表示基因。这种表示法在作业排序问题中显然是不合适的, 因为不仅复杂, 而且不易于操作。最方便的编码方式是直接利用各台机器上的作业排列来表示染色体, 如

$$O_{11} \quad O_{12} \quad \cdots \quad O_{1n} \quad O_{21} \quad O_{22} \quad \cdots \quad O_{2n} \quad \cdots \quad \cdots \quad O_{m1} \quad O_{m2} \quad \cdots \quad O_{mn}$$

其中: $O_{ij} \in \{ J_1, J_2, \dots, J_n \}$ 。每个染色体含有 m 个片段, 每段含 n 个基因, 表示 n 个工件在

m 台机器上的投放顺序。本文算法中, GA 仅对每台机器的首工序优化, 故染色体可简记为

$$O_{11} O_{21} \cdots O_{m1}$$

虽然本文算法能在很大程度上保证所得解的可行性, 但仍会有少量不可行解出现, 这是因为并非任意的工件作首工序都能满足技术约束。然而由于大多数不可行解已通过启发式规则的引入而被消除, 剩下的一小部分很容易处理, 只需在其适应值中加一个惩罚项即可。

(2) 算子设计 由于本文的混合 GA 仅处理每台机器的首工序, 不仅大大简化了染色体的表示, 而且使遗传算子的设计也很容易。在一个染色体中, O_{i1} 可以与某个 O_{j1} 相同, 因此, 在设计杂交算子时不必考虑如何消除“非法染色体”(在同一台机器上, 同一工件出现多次), 只需采用通常的多点杂交(本文采用两点杂交)方法即可。

类似地, 也可直接采用传统的变异算子。本文随机地选取 3 个变异点, 并随机改变其上的工件号。过早收敛是一直困扰遗传算法的主要问题之一, 本文采用了一种自适应变异率方法, 即在 GA 执行过程中, 变异率 P_m 随群体的多样性程度不同而不断改变。若群体多样性程度较高, 则 P_m 减小; 反之, 则增大。群体多样性程度可由下式来衡量

$$\hat{u}f_{\max} - f_{\min}\hat{u}/\hat{f}$$

其中: f_{\max} 与 f_{\min} 分别为群体中诸个体所具有的最大与最小适应值; \hat{f} 为群体的平均适应值。

本文的复制算子在选取个体时, 是根据个体在群体中按适应值大小的排列顺序来加以选择的, 适应值越高的个体, 其序号越大, 被选择的机会也越大。选择概率与个体的序号成比例, 而不是简单地与适应值大小成比例, 这样可以简化适应函数的构造。

邻域搜索法是一种直观、有效的搜索技术, 常被用于求解作业排序问题。其基本思想是从一个可行解(种子)出发, 调整其中的某些工序以生成一组新的解(邻域), 然后检查一下邻域中的解是否比原解有所改进, 若有改进, 则选一个改进解作为新的种子, 继续以上过程, 直至再也找不到更好的解为止。邻域的定义有很多种, 本文采用的是文献[4]中的“单邻域对交换邻域”。

综上所述, 本文的启发式混合遗传算法步骤可简要归纳如下:

- (1) 令 $k=0$, 初始化第 k 代群体 $P(k)$, 并求取 $P(k)$ 中诸个体的 C_{\max} 值;
- (2) 若满足终止条件, 则转步骤(6); 否则, 令 $k=k+1$, 执行步骤(3);
- (3) 对 $P(k-1)$ 中的所有个体, 生成相应的由各台机器首工序组成的染色体, 并由此构成集合 $P'(k-1)$;
- (4) 由 $P'(k-1)$ 通过复制生成 $P'(k)$, 并在 $P'(k)$ 上实施杂交与变异运算;
- (5) 按照启发式调度规则(如 SPT 或 MWKR) 安排各台机器的后续工序, 以生成一个完整的排序, 并由此构成新的群体 $P(k)$, 然后计算 $P(k)$ 中诸个体的 C_{\max} 值, 返回步骤(2);
- (6) 在 $P(k)$ 中选择最佳个体作为种子, 实施邻域搜索。

2 算例分析

本文利用启发式混合遗传算法针对各种不同规模、不同参数的作业排序问题进行了大量的运算, 同时与当前比较常用的排序算法进行了广泛的比较, 如单纯采用调度规则(SPT 与 MWKR^[6])、模拟退火算法(SA)^[7]、邻域搜索算法(NST)^[4]、以及单纯的遗传算法(GA)^[3]等。比较的结果如表 1 所示, 其中, GA-SPT-NST 与 GA-MWKR-NST 分别表示在混合遗传算法中引入 SPT 与 MWKR 规则。所有算例的参数均由计算机随机产生, 对于每

个算例, 本文均运行了 20 次, 每次运行的初始解(或初始群体)均随机生成。

表 1 启发式混合遗传算法与其它算法的比较结果

算 例 $m \times n$	单纯调度规则		SA			NST			GA			GA-SPT-NST			GA-MWKR-NST		
	SPT	MWKR	最好	最差	平均	最好	最差	平均	最好	最差	平均	最好	最差	平均	最好	最差	平均
5×10	1809	1907	1749	1989	1892	1775	1913	1846	1737	1859	1804	1648	1648	1648	1624	1624	1624
10×10	5196	5010	3609	3855	3734	4502	4797	4636	3886	3918	3906	3783	3944	3874	3648	3979	3743
10×10	3223	3207	2409	2517	2459	2293	3021	2722	2266	2282	2274	2220	2382	2269	2235	2293	2284
10×10	5194	5411	3649	3896	3814	5007	5026	5016	3868	3992	3931	3674	4042	3813	3542	3674	3569
10×10	2742	2733	2292	2528	2416	2387	2992	2632	2282	2370	2352	2153	2523	2303	2186	2408	2271
10×15	5294	4575	4104	4286	4181	3637	5028	4577	3698	3761	3734	3583	4084	3710	3458	3702	3543
10×15	4996	4689	4031	4354	4183	3973	4973	4623	3599	3671	3640	3705	4010	3864	3508	3779	3611
15×15	7155	6446	5215	5452	5317	4755	6801	6171	4900	4900	4900	4771	6801	5162	4528	5333	4887
15×15	6124	6227	4373	4704	4538	5736	6011	5874	4212	4315	4280	4268	5508	4841	4049	4679	4241
15×20	8604	8106	7120	7760	7487	7792	8139	7965	5896	6098	6077	5916	7226	6552	5658	6292	5868
15×20	7566	7205	5668	5717	5693	6809	7315	6921	5349	5349	5349	5303	6074	5606	5223	5530	5416
20×20	9876	8870	7939	8303	8066	8213	9270	8742	6184	6419	6279	6765	7928	7310	6229	7592	6844

显而易见, 算法 GA-MWKR-NST 的性能几乎在所有的算例中均遥遥领先, 其所求出的平均次优解仅有 3 例比 GA 略差, 而最好的次优解仅有 1 例不如 GA。但是, 正如前面所预料的, 启发式混合遗传算法的运算效率比单纯遗传算法要高得多, 因而仅需耗费较少的时间即可收敛到满意的次优解。问题规模越大, 混合遗传算法比单纯遗传算法收敛得越快。因此, 本文的算法在处理较大规模的排序问题时, 无论在求解有效性方面还是在计算效率方面均具有较大的优势。

参 考 文 献

- 1 Garey M R, Johnson D S, Sethi R. The complexity of flowshop and job-shop scheduling. Math Opl Res, 1976, 1: 117~129
- 2 Goldberg D E. Genetic algorithms in search, optimization and machine learning. New York: Addison-Wesley, 1989
- 3 Davis L. Handbook of genetic algorithms. New York: van Nostrand Reinhold, 1991
- 4 Siman F. Sequencing and scheduling: an introduction to the mathematics of the job-shop. Chichester: Ellis Horwood Ltd, 1982
- 5 Panwalker S S, Iskander W. A survey of scheduling rules. Ops Res, 1977, 25: 45~61
- 6 Conway R W, Maxwell W L, Miller L W. Theory of scheduling. Reading: Addison-Wesley, 1967
- 7 van Laarhoven P J M, Aarts E H L. Simulated annealing: theory and applications. Dordrecht: D. Reidel Publishing Company, 1987