

智能混合优化策略及其在流水作业调度中的应用

冯远静, 冯祖仁, 彭勤科

(西安交通大学系统工程研究所, 710049, 西安)

摘要: 通过结合蚁群算法(ACO)的并行搜索结构和模拟退火算法(SA)的概率突跳性,提出了一种有效的混合优化策略,并将该策略应用于流水作业调度问题(FSP). 在该策略中,蚁群系统的一个周游路线为模拟退火算法提供了一系列初始解,在每个退火温度上进行抽样准则检验并产生新解,然后更新信息激素;蚁群算法再利用模拟退火算法产生的新解进行并行搜索. 同时,根据此策略构建并实现了针对 FSP 问题求解的具体混合算法. 仿真结果表明,混合算法弥补了 ACO 易陷入局部最优和 SA 搜索效率较低的缺点,增强了全局搜索能力,在求解 FSP 调度问题的性能上也优于其他算法.

关键词: 蚁群算法;模拟退火算法;混合优化算法;流水作业调度

中图分类号: TP278 **文献标识码:** A **文章编号:** 0253-987X(2004)08-0779-04

Intelligent Hybrid Optimization Strategy and Its Application to Flow-Shop Scheduling

Feng Yuanjing, Feng Zuren, Peng Qinke

(Systems Engineering Institute, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: By combining the parallel searching structure of ant colony optimization (ACO) with the probabilistic jumping property of simulated annealing (SA), an effective hybrid optimization strategy was developed, and applied to flow-shop scheduling problems (FSP). In the hybrid strategy, a cycle course of the ant system can provide effective initial solutions for SA, and SA generates new solutions based on the metropolis criterion at each temperature, then the ant system updates pheromone trails and proceeds with parallel searching through reusing the new solutions from SA. Meanwhile, the hybrid algorithm for flow-shop scheduling problems was created and realized on the basis of the hybrid strategy. With some benchmark FSP problems, the simulation results show that the method compensates the deficiency of ACO that is easy to be run into local optimum and SA that has lower efficiency, and strengthen the global search ability. Comparing with other algorithms for solving FSP, the proposed method has better performance.

Keywords: ant colony optimization; simulated annealing; hybrid optimization algorithm; flow-shop scheduling

流水作业调度问题(FSP)是许多实际流水线生产调度问题的简化模型,是一种典型的 NP 难题^[1]. FSP 问题的有效求解方法通常可以分为构造型和改进型方法. 在过去的几年,人们对利用改进型方法求解 FSP 问题进行了大量的研究,比如模拟退火算法(SA)^[2]、禁忌搜索^[3]、遗传算法^[4]和蚁群算法

(ACO)^[5-9]. 在随机搜索算法中,ACO 是一种有效的求解方法,但对于大规模的 FSP 调度问题,它不可避免地陷入了局部最优^[6,7,10]. 相对于 ACO,SA 的概率突跳性可以逃避局部最优^[9],但由于串行搜索结构使其搜索效率较低. 本文通过合理地结合这两种搜索算法的特性,提出了一种有效的混合优化策

略,并通过基于典型 FSP 问题的仿真,证明了所提算法的有效性.

1 智能混合优化算法

1.1 混合优化算法的基本思想

ACO 是模拟蚁群从蚁巢到目的地寻找最短路径的并行搜索算法. SA 是模拟物理退火过程的一种算法,通过赋予搜索过程一种时变且最终趋于 0 的概率突跳性,从而有效避免了局部极小. 如果两者相结合,其性能的互补性主要体现在以下几个方面.

(1)结合性的搜索机制. ACO 和 SA 均是基于概率机制的随机优化算法,其中:ACO 是基于群体搜索的算法,收敛速度很快,但由于信息激素更新能力有限,所以经常会出现“早熟”现象;SA 通过抽样稳定准则产生新解,并由退火温度控制搜索过程,从而有效避免了局部最优^[11]. 两者在机制上的结合有利于丰富优化中的搜索行为,同时可增强全局和局部意义下的搜索能力和效率.

(2)互补性的优化结构. SA 采用串行优化结构,ACO 采用并行搜索,两者结合可使 SA 成为并行 SA,从而提高优化性能. 同时,SA 的概率突跳性可增强蚁群的信息激素更新能力.

(3)结合性的历史信息搜索操作. SA 在状态产生和接受算子时仅留一个解,缺乏冗余和历史信息搜索^[11],蚁群的信息激素保留了历史信息搜索. 两者结合有利于丰富优化过程中的邻域搜索结构,增强全空间的搜索能力.

(4)削弱参数选择的苛刻性. SA 的优化行为对退温历程有很强的依赖性,而理论上的全局收敛对退温策略的限制条件很苛刻. ACO 的搜索过程不需要进行人工的调整,具有很好的鲁棒性. 同时,SA 还可以平衡蚁群搜索过程中知识利用与探索之间的关系.

1.2 算法描述

基于上述的基本思想,本文在 ACO 的框架基础上引入了 SA,提出了一种有效的混合优化算法 (ACSA),该算法的流程如图 1 所示.

从图中可以看到,ACO 为 SA 提供了一系列的初始解,SA 的邻域搜索能力进一步改善解,然后 ACO 利用 SA 产生的新解再进行并行搜索. 所以,ACSA 结合了两者的特点,使各自的搜索能力得到补充,弱点得以弥补.

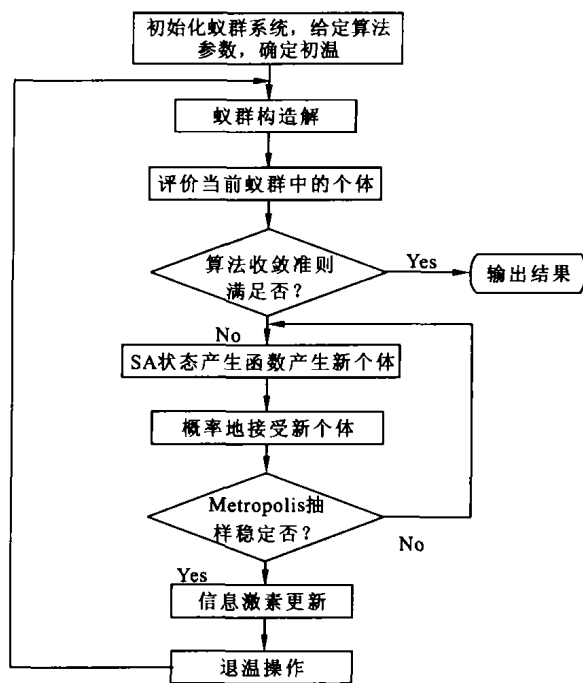


图 1 混合优化算法流程图

2 基于混合优化算法的 FSP 求解

2.1 解的构造

假设一个置换 FSP 调度问题研究 m 台机器上 n 个工件的流水加工过程, t_{ij} 为工件 i 在机器 j 上的加工时间, θ_{ik}^j 为机器 k 上完成加工工件 i 到要加工下一个工件 j 时所需的准备时间, T_i 为工件 i 的加工时间, D_i 为工件 i 的计划完成时间. 不失一般性,另设各工件按 $1 \sim m$ 个机器的顺序进行加工,令 $\pi = (\sigma_1, \sigma_2, \dots, \sigma_n)$ 为所有工件的一个排序,则调度的目标是寻找一个排序,使工件的完成时间最短.

在 ACSA 中,蚂蚁的信息激素采用与解的属性有关的数值 τ_{ij} 来表示,即工件 i 在加工序列中的第 j 个位置的期望程度. 每个人工蚁从一个虚构的起始结点(初始作业)出发,然后依次选择后续的作业,一步一步地构造出一条完全路径(解序列). 在构造路径的每一步,蚂蚁 k 从结点 i 移动到相邻结点 s , 结点 s 表示将被插入处理序列 $\pi' = (\sigma_1, \sigma_2, \dots, \sigma_i)$ 的下一个作业, s 的选择按照伪随机比例状态迁移规则进行^[12],即

$$s = \begin{cases} \arg \max_{i \in N_i^k} ([\tau_{is}(t)]^\alpha [\eta_{is}(t)]^\beta), & q \leq q_0 \\ s', & q > q_0 \end{cases} \quad (1)$$

蚁群以概率 $q_0 \in [0, 1]$ 选择结点 $\arg \max_{i \in N_i^k} ([\tau_{is}(t)]^\alpha \cdot [\eta_{is}(t)]^\beta)$, 以概率 $1 - q_0$ 并按概率

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha [\eta_{il}(t)]^\beta}, & j \in N_i^k \\ 0, & j \notin N_i^k \end{cases} \quad (2)$$

选择 s' . 在搜索时, 参数 q_0 决定了蚁群的知识利用与探索两者的权重. 一味的知识利用将使搜索很快地陷入局部最优, 而过多的探索将影响算法的性能. 式(2)中的 $\eta_{ij}(t)$ 是与目标函数有关的启发式信息, α, β 分别是控制信息激素和启发式信息值在概率 $p_{ij}^k(t)$ 中的权重参数.

2.2 初始温度及温度更新函数

m 个人工蚁完成第一次周游后, 确定周游最优路径(一次周游路径中目标函数值最小的路径)及其完成时间 c_{best} , 和周游最差路径(一次周游路径中目标函数值最大的路径)及其完成时间 c_{worst} . 确定初始温度 $t_0 = -(c_{\text{worst}} - c_{\text{best}}) / \ln(p_0)$, 其中的 $p_0 \in [0, 1]$ 为初始接受概率. 退温函数采用指数退温策略 $t_k = \lambda t_{k-1}$, $\lambda \in (0, 1)$ 为退温速率. 这种退温策略被认为是一种有效的策略^[13], 可较好地折衷、兼顾优化质量和时间性能.

2.3 状态产生函数和状态接受函数

在流水作业调度问题中, 状态的产生可以通过一个置换来实现, 而置换的邻域可以表示为一系列作业的移动操作. 本文将互换操作^[13]作为算法的状态产生函数.

为了使搜索过程能克服局部极小, 并满足 SA 算法的对称条件, 在结合互换操作状态产生函数的基础上, 用 Metropolis 抽样稳定准则作为接受新状态的条件.

人工蚁完成一次周游以后, 为 SA 的邻域搜索提供了初始解. 邻域搜索算法描述如下.

Step 1: 在温度 t_n , 选择人工蚁 k 的周游路径 π_k .

Step 2: 通过互换操作随机产生一个新的邻域

解 $\tilde{\pi}_k$, 并计算 $\pi_k, \tilde{\pi}_k$ 的目标函数值 $c(\pi_k)$ 和 $c(\tilde{\pi}_k)$.

Step 3: 如果 $\min\{1, \exp[-(c(\tilde{\pi}_k) - c(\pi_k)) / t_n]\} \geq \text{random}[0, 1]$, $\pi_k = \tilde{\pi}_k$; 否则, 转 Step 2.

Step 4: 如果所有的邻域解都不能改善当前解 π_k , 或者搜索的次数达到某一设定值, 则停止搜索; 否则, 转 Step 2.

2.4 信息激素的更新

在算法的每一次迭代中, 当整个蚁群中的所有人工蚁完成邻域搜索并构造其问题时, 每个人工蚁 k 将在经过的路径连接上积累一定量的信息激素 $\Delta\tau_{ij}^k = 1/c_{ij}^k(N)$, 其中的 $c_{ij}^k(N)$ 是蚂蚁 k 在第 N 次迭代时经过路径 $\psi^k(N)$ 完成的时间. 信息激素依据

$$\tau_{ij}^k(N) = \rho \tau_{ij}^k(N-1) + \Delta\tau_{ij}^k(N) \quad (3)$$

更新. 其中, $\rho \in (0, 1)$ 表示信息激素的保留系数, $1 - \rho$ 表示信息激素的蒸发率.

2.5 循环终止准则

ACSA 利用两种循环终止准则, 其中: 抽样稳定准则可判定一个温度下的算法搜索行为, 也是 SA 切换到 ACO 的条件. 算法终止原则可判定优化性能变化趋势和最终优化性能.

3 仿真结果及分析

为了验证 ACSA 的有效性, 选择了不同规模、典型的置换 FSP 问题进行测试. 典型的置换 FSP 问题采用 Car、Rec 和 TA 标准测试集的部分数据^[5], 其中的 Car1、Car3、Car5 和 Car7 作为简单调度问题, 其余的作为调度难题进行求解. 设置参数 $m = 10$, $p_0 = 0.1$, $\lambda = 0.9$, $\rho = 0.9$, $q_0 = 0.9$, 并且每个实例均随机运行 20 次. 用 C 语言实现上述算法, 运行环境为 Pentium IV 2.4 GHz/256 MB RAM.

首先, 将 ACSA 分别与 ACO 和 SA 的优化性能进行比较, 仿真计算结果见表 1. 从表 1 可以看

表 1 ACSA、ACO 和 SA 的优化性能比较

标准测试集	n, m	c^{**}	ACSA			SA			ACO		
			c^*	$e/\%$	t/s	c^*	$e/\%$	t/s	c^*	$e/\%$	t/s
Car1	11, 5	7 038	7 038	0	0.35	7 038	0	0.13	7 038	1.345	0.11
Car3	12, 5	7 312	7 312	0	1.25	7 312	0	0.33	7 312	2.123	0.12
Car5	10, 6	7 720	7 720	0	1.15	7 720	0	0.28	7 720	0	0.10
Car7	7, 7	6 590	6 590	0	1.12	6 590	0	0.23	6 590	0	0.09
Rec01	20, 5	1 247	1 247	1.567	5.56	1 259	6.323	2.34	1 278	2.462	0.71
Rec07	20, 10	1 566	1 576	1.533	7.23	1 578	9.453	3.24	1 596	4.253	1.18
Rec13	20, 15	1 930	1 930	1.045	8.86	1 939	4.245	4.56	1 986	5.678	1.34
Rec19	30, 10	2 093	2 106	2.563	14.36	2 122	3.245	8.78	2 289	2.732	2.67
Rec25	30, 15	2 513	2 535	3.632	21.25	2 625	7.532	14.67	2 675	5.756	3.56
Rec31	50, 10	3 045	3 126	1.067	59.25	3 286	5.482	18.45	3 361	6.734	4.67
Rec37	75, 20	4 951	5 024	6.345	145.34	5 122	8.432	56.10	5 232	11.563	12.41

注: n, m 表示问题的规模; c^{**} 为问题的最优解; c^* 为本算法的最优解; e 表示 20 次运行结果与 c^{**} 的相对误差; t 表示平均 CPU 时间.

出,对于简单实例,3种算法都能很好地求解问题,对于调度难题,ACSA的优化性能更好.从算法得到的问题最优值可以看出,ACSA的全局搜索能力得到了较大的改进,从20次运行结果得到的平均相对误差值显示出了ACSA的鲁棒性.但是,ACSA的时间性能不如ACO和SA的.由于ACO本身内在的并行性质,可使得利用ACSA并行实现来提高

算法的速度和运行效率成为可能,这将是我們下一步的研究工作.

另外,我们也将ACSA在求解TA标准测试问题^[3]时的优化性能与其他几种算法进行了比较,结果见表2,其中的数据全都来源于文献[4].比较结果表明,ACSA的优化性能更好或者至少与其他算法相当.

表2 ACSA和其他算法的优化性能比较

标准测试集	m, n	平均偏差					
		NEH	NEH+1S	SAOP	MMAS	MD	ACSA
Ta001-010	20, 5	3.663	1.923	1.061	0.408	0.730	0.208
Ta011-020	20, 10	4.601	2.453	1.462	0.591	0.955	0.397
Ta021-030	20, 20	3.731	2.417	1.116	0.410	0.628	0.409
Ta031-040	50, 5	0.727	0.258	0.597	0.145	0.255	0.148
Ta041-050	50, 10	5.073	3.438	3.012	2.193	2.873	1.678
Ta051-060	50, 20	5.971	4.267	3.147	2.475	3.138	2.145
Ta061-070	100, 5	0.527	0.324	0.509	0.196	0.219	0.134
Ta071-080	100, 10	2.215	1.250	1.719	0.928	1.226	0.873
Ta081-090	100, 20	4.275	2.942	3.249	2.238	3.229	1.876

注:NEH为启发式算法;NEH+1S为NEH加邻域搜索;MMAS为max-min蚁群算法;MD为多重下降法;SAOP为改进的SA.

4 结 论

ACSA是建立在蚁群算法结构基础上的一种并行搜索算法.利用模拟退火算法的Metropolis准则折衷处理蚁群搜索的知识利用与探索之间的平衡,能增强算法在解空间中的探索能力和效率.

ACSA具有较好的全局优化度,其初值鲁棒性高,优化结果可靠.

对于复杂优化问题,如FSP的NP难题,单一机制的优化算法很难实现全局优化且效率低.结合多种优化机制和邻域搜索结构的ACSA是提高全局优化能力和鲁棒性的有效方法,并可以在一定程度上放宽单一算法在参数选择上的苛刻性要求.

参考文献:

- [1] Garey E L, Johnson D S, Sethi R. The complexity of flow-shop and job-shop scheduling [J]. Mathematics of Operations Research, 1976, 1(1):117-129.
- [2] Osman I, Potts C. Simulated annealing for permutation flow-shop scheduling [J]. Omega, 1989, 17(6): 551-557.
- [3] Liaw C F. An efficient tabu search approach for the two-machine preemptive open shop scheduling problem [J]. Computers & Operations Research, 2003, 30(14):2 081-2 095.
- [4] Taillard E. Some efficient heuristic methods for flow-shop sequencing [J]. European Journal of Operational

Research, 1990, 47(1):65-74.

- [5] 王 凌著. 车间调度及其遗传算法 [M]. 北京:清华大学出版社, 2003.
- [6] Stutzle T. An ant approach to the flow shop problem [A]. European Congress on Intelligent Techniques and Soft Computing, Aachen, Germany, 1998.
- [7] Dorigo M, Caro G D. Ant colony optimization: a new meta-heuristic [A]. Int Conf on Evolutionary Computation, Piscataway, USA, 1999.
- [8] 王 凌, 郑大钟. 一种GASA混合优化策略 [J]. 控制理论与应用, 2001, 18(4):552-554.
- [9] Dorigo M, Gambardella L M. Ant colony system: a cooperative learning approach to the traveling salesman problem [J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1):53-66.
- [10] Dorigo M, Manjezzo V, Colomi A. Ant system: optimization by a colony of cooperating agents [J]. IEEE Transaction on Systems, Man & Cybernetics, B, 1996, 26(1):29-41.
- [11] Stutzle T, Hoos H H. The max-min ant system and local search for the traveling salesman problem [A]. IEEE 4th International Conference on Evolutionary Computation, Piscataway, USA, 1997.
- [12] Stutzle T, Hoos H H. Max-min ant system [J]. Future Generation Computer Systems, 2000, 16(8):889-914.
- [13] Hajek B. Cooling schedules for optimal annealing [J]. Mathematics of Operations Research, 1988, 13(2): 311-329.

(编辑 苗 凌)