多目标满载装卸货问题的蚁群算法研究

徐为明
XU Wei-ming

上海商学院 管理学院,上海 200235
Institute of Management,Shanghai Business School,Shanghai 200235,China
E-mail:xwm72100@163.com

XU Wei-ming.Research for multi-objective full load pickup and delivery problem based on ant colony algorithm.Computer Engineering and Applications,2009,45(31):227-229.

Abstract: Full Load Pickup and Delivery Problem(FLPDP) is an important combinatorial optimization problem existed extensively in the transportation domain.In order to solve practical multiple-objective full load pickup and delivery problem,the bi-level Max-Min Ant algorithm(MMAS) is proposed.With the positive feedback and parallelism of the Max-Min ant colony algorithm,two objectives of the problem are optimized simultaneously through the pheromone exchanging between the two ant colonies.The simulative computational results demonstrate that the proposed algorithm is able to procedure feasible results for the large-scale problem.

Key words: Max-Min Ant algorithm (MMAS);multi-objective;Full Load Pickup and Delivery Problem(FLPDP)

摘　要:满载装卸货问题是广泛存在于物流运输领域的重要组合优化难题。为了有效求解实际情况下多目标的满载协同运输问题,设计了双层最大最小蚁群算法。利用蚁群算法的正反馈和并行性,通过不同层次蚁群之间的信息素传递,实现对问题的两个优化目标同时优化。通过实验表明了该算法可行而有效。

关键词:最大最小蚁群算法;多目标;满载装卸货问题
DOI:10.3778/j.issn.1002-8331.2009.31.068　**文章编号**:1002-8331(2009)31-0227-03　**文献标识码**:A　**中图分类号**:TP301.6

1 引言

车辆路径和调度问题(Vehicle Routing Problem,VRP)堪称物流运输环节中最重要的运作规划问题[1]。选择恰当的车辆调度方法,可以加快对客户需求的响应速度,提高服务质量,增强客户对物流环节的满意度,降低物流服务商的运作成本。车辆调度问题自提出后,由于其重要的现实意义,涌现出了一大批求解的精确算法和启发式算法,如节约法、Sweep 算法、2-opt 和 3-opt 交换算法、遗传算法、禁忌搜索算法等。装卸货运输问题(Pickup and Delivery Problem)是一类重要而特殊的车辆路径和调度问题,在物流运输领域具有广泛的应用[2-4]。在此运输方式下,车辆自车场出发,行驶到某装货点装载货物,将货物运输到此装载点对应的卸货点,进行卸货运作,再进行下一次运输,车辆最终必须返回出发的车场。由于满载运输(Full load Transportation)是常用的一种运输方式,该文主要考虑满载运输方式下的装卸货运输问题。在满载装卸货运输问题中,车辆在某点装载了货物后,必须将货物直接送至目的地,即在完成此次运输任务之前,不允许穿插服务其他客户,客户对为其服务的车辆资源具有独占性。满载装卸货运输问题是复杂的优化问题,求解非常困难,对之探讨的文献很少,现有的算法也多为利用简单的启发式算法进行近似求解。蚁群算法是一种新型的模拟进化算法,在诸如 Job-shop 问题、TSP 中都表现出了非常优良的特性。文章针对一类多目标的满载装卸货运输问题,在分析问题性质的基础上,设计了双层最大-最小蚁群算法(MMAS)[5],对问题加以有效解决。

2 问题描述及模型

基于图论理论,多目标满载装卸货运输问题可以概述如下:在有向平面图 $G=(V,A)$中,顶点集 $V=\{V_0,V_1,\cdots,V_n,V_{n+1},\cdots,V_{n+n}\}$。节点 v_0 表示车场,节点$\{V_1,V_2,\cdots,V_n\}$表示 n 个装载点,节点$\{V_{n+1},V_{n+2},\cdots,V_{n+n}\}$表示各个装载点各自对应的卸货点。车辆需要自 v_0 出发,空车行驶到某任务的装运点 $V_i(1\leqslant i\leqslant n)$,装载货物后行驶到点 V_{i+n} 卸货,之后车辆直接返回 v_0,或者前往下一个任务的装运点,继续装运和运输。车辆最终必须回到车场 v_0。在上述条件下,车辆自 V_i 装载货物后必须直接行驶至 V_{i+n} 点,因此执行 V_i 至 V_{i+n} 的运输路径必然为两点之间的最短弧。由于平面图中三角距离不等式成立,当 V_i 和 V_{i+n} 之间具有弧直接连接时,此两点间的任务执行可以转化成对弧(V_i,V_{i+n})的覆盖;否则,可以通过 Dijkstra 算法求得 V_i 和 V_{i+n} 之间最短距离的有向弧序列$(V_i,V_{i+k},\cdots,V_{i+l},V_{i+n})$,运输任务可转换成对此有向弧序列覆盖。为了简化运算,可合并有向弧序列,将此有

作者简介:徐为明(1971-),男,经济学硕士,讲师,主要研究领域为主要研究领域是物流管理、零售业的经营管理、零售企业的采购管理、连锁经营管理等。
收稿日期:2008-06-13　**修回日期**:2008-10-16

向弧序列转换定义为有向弧(V_i, V_{i+n})。

考虑到实际情况下对车辆司机等人员具有每天工作时间的限制，此限制条件可转换为每个车辆的运输路线长度限制，即车辆从车场出发至返回车场的总路径长度小于定值 H。运输的成本由两部分构成：每派出一辆车进行运输的固定成本和车辆运输行驶成本。后者可转换为车辆的行驶距离，而派出一辆车的固定成本远大于车辆行驶距离的增加成本。

因此车辆满载装卸货运输问题的目标是在有向图 G 中确定闭回路集合，每个闭回路均经过车场点 v_0，且长度不超过 H。所有运输有向弧（集）均被闭回路覆盖。闭回路的数目对应使用车辆的数目，闭回路的长度对应于车辆行驶距离。由于派出一辆车的费用远大于车辆路程增加的费用，因此优化闭回路数目为第一目标，优化闭回路的总长度为第二目标。第二目标服从于第一目标，即闭回路数较小的解必定优于闭回路数目多的解，只有在第一目标等同的情况下才比较第二目标的质量。

3　双层最大–最小蚁群算法

由于基本的 PDP 是 NP–hard 问题[6]，上述车辆满载装卸货运输问题在 PDP 的基础上增加了车辆行驶距离的限制，且具有两个优化目标，显然同样具有 NP–hard 性质。目前采用精确算法只能求解小规模的装卸货运输问题，装载点和卸货点数目不超过 100。因此，采用恰当有效的启发式算法是解决此类问题的主要手段和研究方向。

蚁群算法是一种新型的模拟进化算法，在 20 世纪 90 年代首先提出[7]，称之为蚁群系统（Ant Colony System）。算法模拟真实蚁群的觅食行为：蚂蚁在觅食运动过程中，能够在它所经过的路径上留下信息素，而蚂蚁在运动过程中能够感知信息素而进行个体之间信息传递，并以此指导自己的运动方向。因此由大量蚂蚁组成的蚁群集体行为便表现出一种信息正反馈现象。某一路径上单位时间走过的蚂蚁越多，表明该路线的可用性越好，则后来者选择该路径的概率就越大。蚂蚁个体之间就是通过这种信息的交流寻找最优的到达食物源线路。通过对 TSP 和 JSP 等经典问题运用蚂蚁算法的求解证明，蚁群算法具有实现简单、正反馈、分布式等优点，同时其具有很好的全局优化能力和本质上的并行性，相对遗传算法、模拟退火算法等早期进化算法具备更强的鲁棒性，求解时间短且易于计算机实现[8]，特别适合于对离散优化问题的解空间进行多点非确定性搜索。由于在智能算法和进化算法中加速收敛和防止早熟、停滞现象是普遍存在的矛盾，基本蚁群算法在此方面被加以改进并产生了最大–最小蚁群系统（MMAS）。MMAS 限制了信息素的取值范围，在算法中采用了轨迹平滑机制，能够有效避免在搜索中算法过早收敛于某一次优的解。该文采用 MMAS 求解多目标满载装卸货问题，并结合该问题的性质，对基本的 MMAS 进行了改进。由于问题具有两个优化目标，因此针对每一个优化目标各自设计了独立的 MMAS 进行优化。第一个 MMAS 优化问题的首要目标，即减少运输车辆数目；第二个 MMAS 在前者的基础上，优化减少车辆的行驶距离。两个 MMAS 之间通过信息素进行信息的传递，共同完成问题的求解。

3.1　最大–最小蚁群算法及改进

MMAS 中每一只蚂蚁都是一个独立的智能体，蚂蚁通过正反馈机制和通信机制，依据当前路段上的信息素的水平，随机选择下一步的路线。针对该问题改进后的 MMAS 描述如下：

（1）初始化各控制参数，设定最大迭代次数和时间。

（2）在起始点 v_0 安置 m 只蚂蚁，由于初始不能确定需要的车辆数目，m 可取较大值。

（3）初始设定所有路径上的信息素轨迹强度为常数 τ_{ij}。

（4）启动一只蚂蚁 k，确定允许访问的运输任务有向弧集合 allowedk。allowedk 受两个约束条件限制：设蚂蚁 k 目前位于 s 点，首先是算法记录此次遍历已经访问过的运输任务有向弧，不允许该蚂蚁重复访问。同时算法记录蚂蚁 k 已经遍历的路径总距离 L_{ks}。设运输任务有向弧(i,j)的长度为 L_{ij}，s 点到 i 的最小距离为 L_{si}，j 点到车场 v_0 的最小距离为 L_{j0}，若满足：$L_{ks}+L_{si}+L_{ij}+L_{j0}\leqslant H$，则运输任务有向弧$(i,j)\in allowed_k$。

（5）利用启发式规则确定蚂蚁 k 在点 s 对运输任务有向弧$(i,j)\in allowed_k$ 上的固有启发信息 $\eta_{sij}=L_{ij}/(L_{si}+L_{j0}+L_{ij})$。

（6）计算一只蚂蚁 k 在 s 节点处选择运输任务有向弧(i,j)的转移概率：

$$p_{sij}^{k}=\begin{cases}\dfrac{\tau_{sij}^{\alpha}(k)\eta_{sij}^{\beta}(k)}{\sum_{s\in allowed_k}\tau_{sij}^{\alpha}(k)\eta_{sij}^{\beta}(k)}, & j\in allowed_k\\ 0, & \text{otherwise}\end{cases}$$

上式中 α 和 β 为两个参数，α 代表运输任务有向弧(i,j)上信息素 τ_{ij} 对蚂蚁选择路径的重要性，β 反映了路径固有启发信息 η_{ij} 对蚂蚁的启发作用。

（7）确定蚂蚁 k 在 s 节点处选择的运输任务有向弧(i,j)：

$$j=\begin{cases}\arg\max\limits_{s\in allowed_k}\max\{_{sij}^{\alpha}(k)\eta_{sij}^{\beta}(k)\}, & y<q_0\\ \text{依据转移概率 } p_{sij}^{k}\text{ 选择}, & y\geqslant q_0\end{cases}$$

上式表示选择下一条有向弧的可能性是根据伪随机比例原则确定的，即采用了确定性选择和随机选择相结合的选择策略。上式中 y 为蚂蚁每一步路径选择时的随机数且 $y\in[0,1)$，q_0 为常数且 $q_0\in[0,1]$。当算法进化到一定代数后，进化方向已经基本确定时，此方法可以较好地缩小最好和最差路径上的信息量的差距，并且适当加大随机选择的概率，以利于更加完全搜索解空间，从而可以有效地克服基本蚁群算法的不足。实际计算证实，$q_0\in[0.8,0.9]$比较合适。

（8）当蚂蚁返 k 回到点 v_0 后，判断所有需要覆盖的运输任务是否均被蚂蚁访问。若不满足，转到步骤（4），启动新的蚂蚁继续访问运输路线。

（9）计算每只蚂蚁的搜索路径，计算本次路径的总长度 L_k。

（10）更新本次访问路径的信息素，可以描述为：

$$\tau_{ij}(new)=\rho\tau_{ij}(old)+\Delta\tau_{ij}$$

$$\Delta\tau_{ij}=\begin{cases}\dfrac{Q}{L_k}, & \text{本次遍历中蚂蚁经过弧}(i,j)\\ 0, & \text{else}\end{cases}$$

式中 $\Delta\tau_{ij}$ 表示本次循环中路段(i,j)的信息素增量，ρ 为$(0,1)$之间的常数，表示在每次循环时信息素的挥发率，$(1-\rho)$为信息素轨迹的衰减系数。Q 为常量，表示蚂蚁完成一次完整的路径搜索所释放的信息素总量。

（11）设 L_{best} 为目前蚂蚁寻找到的最优路线，若本次访问路线 L_k 较之 L_{best} 更加优化，则 $L_{best}=L_k$。

（12）更新 L_{best} 式方案中的各条路线信息素。

由于算法中每次迭代过程采用最好的蚂蚁进行更新，这使得某些路径上的信息素逐渐挥发殆尽，从而使得蚂蚁聚集选择

信息素最高的路径，致使次路径不断重复使用，使得路线的搜索陷入停顿，容易引起算法的早熟。该算法定义了一个取值区间[$\tau_{\min}$，$\tau_{\max}$]，把任意弧上的信息素限制在此区间内。如果 $\tau_{ij}<\tau_{\min}$，设定 $\tau_{ij}=\tau_{\min}$；如果 $\tau_{ij}>\tau_{\max}$，设定 $\tau_{ij}=\tau_{\max}$。由于每次迭代后，最优路径上信息素增加和 Q/L_K 成正比例，该文设定 $\tau_{\max}=Q/[(1-\rho)*L_{best}]$，$\tau_{\min}=1/20*\tau_{\max}$。

（13）判断迭代次数是否等于最大迭代次数，或者是计算时间达到限定时间。若是，则流程结束，输出最优解；若否，跳回步骤（2），重复进行上述计算。

3.2 双层蚁群算法设计

由于上述车辆满载装卸货运输问题具有两个优化目标，对于多目标的算法，通常的处理方法是对不同的目标进行线形加权或者是将某些目标转化成约束条件[9]。但是利用线形加权法时确定系数权重非常困难。该文借鉴文献[10]的方法，通过构造两层蚁群进行分层次优化。每个蚁群用以优化问题的一个目标。第一个层次的蚁群 Ant-Vn 优化车辆数目，尽量减少调度方案使用的车辆数目；同时第二个层次的蚁群 Ant-Ds 优化所有车辆的最小行驶距离。蚁群 Ant-Vn 和 Ant-Ds 均按照上述 MMAS 的基本流程进行设计。而两层蚁群之间需要进行蚁群间的信息素交换以交换各自的寻优信息。由于优化车辆数目是问题的第一目标，当第一层次的蚁群求解出了更好的第一目标，则停止第二层次的蚁群，并按照当前最优解重新启动第二层次的蚁群；当第二层次蚁群发现了更好的第一目标，同样按照发现的最优解重新启动第一层次的蚁群；当第二层次蚁群仅发现了更好的第二优化目标，则更新第一层次蚁群的信息素，以达到信息的交换和影响。两层蚁群算法的流程具体如下：

（1）采用节约法对车辆满载装卸货运输问题生成一个初始解；

（2）设定 ψ_{best} 代表两层蚁群算法的整体最优解。由初始解确定 ψ_{best}；

（3）设 ψ_{best} 的车辆数为 V_{best}；ψ_{best} 中车辆的行程为 D_{best}；

（4）按照 ψ_{best} 设定 Ant-Vn 的初始信息素；

（5）按照 ψ_{best} 设定 Ant-Ds 的初始信息素；

（6）Ant-Vn 开始迭代运算；

（7）Ant-Ds 开始迭代运算；

（8）当 Ant-Vn 产生更优的解，即优化了 V_{best}，依据 Ant-Vn 优化结果更新 ψ_{best}；判定算法是否达到了时间的限制；如果达到时间限制，转到步骤（18）；

（9）依据 ψ_{best} 更新 Ant-Vn 信息素；

（10）停止 Ant-Ds，设定 ψ_{best} 为 Ant-Ds 的初始解，按照 ψ_{best} 设定 Ant-Ds 的初始解和初始信息素，转到步骤（7）；

（11）当 Ant-Ds 优化了 D_{best}，判定 Ant-Ds 当前的解是否优化了 V_{best}，若否，转到步骤（16）；

（12）Ant-Ds 的当前解优化了 V_{best}，设定 ψ_{best} 为 Ant-Ds 的当前解；

（13）判定算法是否达到了时间的限制；如果达到时间限制，转到步骤（18）；

（14）停止 Ant-Vn，按照 ψ_{best} 初始化 Ant-Vn 信息素；

（15）按照 ψ_{best} 更新 Ant-Ds 的信息素，转到步骤（6）；

（16）判定 Ant-Ds 当前解车辆数是否大于 V_{best}，若是，停止 Ant-Ds，按照 ψ_{best} 设定 Ant-Ds 的初始解和初始信息素，转到步骤（7）；

（17）设定 ψ_{best} 为 Ant-Ds 的当前解；按照 ψ_{best} 更新 Ant-Vn 和 Ant-Ds 的信息素；转到步骤（6）；

（18）结束算法，输出目前的 ψ_{best}。

4 算例结果以及分析

为了分析文章提出的双层 MMAS 算法的有效性和可行性，采用两组大规模随机产生的实验数据，进行实验计算验证。算法程序使用 C 语言编写，在 Inter Pentium1.6 GHz，1 GMB 内存的 PC 上进行测试演算。两组算例的客户点均为 200 个，均匀分布在半径为 8 的圆形区域内。在两组测试数据中随机选取两个客户点作为车辆的装载点和卸货点，运输任务数目分别为 200 和 500。车场点为圆周上的随机产生的一个点。算法采用的控制参数为：$\alpha=2$；$\beta=3$；$q_0=0.85$；$\rho=0.75$；迭代次数设定为 1 500 次，最终结果取三次重复计算结果均值。由于目前针对多目标满载装卸货问题而提出的有效算法非常少，文章将和文献[11]提出的两阶段快速算法进行比较分析。表 1 显示了对两组算例的计算对比结果。

表 1　两组大规模算例计算结果

路线数目	H	双层 MMAS 算法		两阶段快速算法	
		车辆数目	行驶距离	车辆数目	行驶距离
200	1 000	2	1 940.9	2	1 930.6
	800	3	1 974.1	4	1 975.1
	600	4	2 000.0	4	1 975.1
	400	6	2 059.6	8	1 986.3
	200	12	2 189.3	15	2 005.5
	150	16	2 237.1	21	2 112.9
	100	27	2 447.6	38	2 286.3
	80	37	2 600.7	51	2 475.1
500	2 000	3	4 720.8	3	4 705.6
	1 500	4	4 731.2	4	4 712.5
	1 000	5	4 772.6	4	4 712.5
	800	7	4 821.6	7	4 820.7
	600	9	4 882.1	11	4 830.5
	400	13	4 961.7	18	4 861.2
	200	28	5 259.2	36	4 980.1
	150	39	5 438.3	57	5 217.0
	100	64	5 830.7	81	5 674.8

通过对两组大规模的算例的计算表明，该文提出的双层 MMAS 算法求解多目标的车辆满载装卸货运输问题可以得到优良的结果。分析后可得到以下具体结论：

（1）在相同的运输路线条件下，随着车辆最大运输距离 H 增加，双层 MMAS 算法和两阶段快速算法得到的车辆使用数目和车辆行驶总距离均随之减少。

（2）当车辆最大运输距离 H 取值较大时，双层 MMAS 算法和两阶段快速算法得到的车辆使用数目非常接近，而两阶段快速算法计算得到的车辆行驶总距离略优。

（3）随着车辆最大运输距离 H 取值减小，双层 MMAS 算法优化结果的第一优化目标，即车辆数目，小于两阶段快速算法的结果；而后者对问题第二目标的优化结果优于双层 MMAS 算法。由于第一优化目标的重要度高于第二优化目标，所以双层 MMAS 算法优化的结果优于两阶段快速算法。

综合考虑，该文设计采用的双层 MMAS 算法在求解大规模多目标的车辆满载装卸货运输问题时，相对已有的两阶段快速算法是更加有效的。

（下转 244 页）