

面向第 4 方物流的多代理人作业整合优化算法^①

王 勇¹, 吴志勇¹, 陈修素², 吴金奇¹

(1. 重庆大学经济与工商管理学院, 重庆 400044 2. 重庆工商大学数学与统计学院, 重庆 400067)

摘要: 物流作业整合是现代物流方法中减少物流成本的主要手段. 在综合考虑代理商选择和线路优化两问题的基础上, 建立了基于图状结构的面向第 4 方物流的多代理人作业整合优化模型. 为了避免将代理商选择和线路优化作为两个互相分离的子问题来处理, 提出了求解它的两层邻域搜索算法. 第 1 层采用了转移、交换和环形移动 3 种移动策略求解作业在代理商之间的分配, 而在第 2 层提出了路合并、路生成替换策略, 形成了适于该问题的优化算法. 随机产生 20 例算例, 将两层邻域搜索算法的计算结果与基于 k-最短路的枚举算法的计算结果进行比较, 说明了该算法的可行性和有效性.

关键词: 第 4 方物流; 多代理人; 作业整合; 邻域搜索算法

中图分类号: F224.3; O22 **文献标识码:** A **文章编号:** 1007-9807(2009)02-0105-12

0 引 言

物流与供应链已成为当前管理科学的研究热点^[1,2]. 随着第 4 方物流的发展, 物流作业整合已成为现代物流方法中减少物流成本的主要手段. 探讨第 4 方物流整合商如何针对多个代理商进行优化分配和对多个物流作业进行作业路线整合, 已成为管理科学界十分迫切的任务. 为了深入研究有效的物流整合方法, 学术界已开始研究面向第 4 方物流的多代理人作业整合优化模型.

Lau^[3] 基于拟指派问题对第 4 方物流如何选择代理商进行了研究, 探讨了求解该问题的贪婪算法 (greedy algorithm), 但该文提出的组合优化模型比较简单, 没有考虑到具体作业路线安排, 因而也无法考虑路线整合问题; 陈建清^[4] 提出了第 4 方物流运作决策支持系统框架, 他在文献 [5] 中建立了赋予多维权的有向图模型, 并借鉴图论中的 Dijkstra 算法来解决上述模型, 但没有考虑作业整合带来的价格折扣因素, 因而难以描述物流作业整合引起的成本下降. Virginie^[6] 研究了基于

价格折扣的作业整合模型, 并讨论了该模型的启发式算法, 但是该模型没有涉及到代理商的选择, 是对该问题的一种简化. 文献 [7] 把问题分解为路线优化和承运商选择两个相互分离的子问题. 在最短路线计算完成后, 用资源分配模型进行承运商选择. 但该文存在以下问题: (1) 把问题分成两个相互分离的子问题来建模和计算必然带来最优性的损失, 而且该模型没有考虑按批量整合所节约的费用. 事实上因为各类整合可以节约费用, 走“绕路”若能整合其它作业, 有可能比模型计算的最优路线更节约, 即该方法没有考虑批量整合; (2) 在代理商能力有限即运力紧张时, 方案不具有最优性. (3) 该文没有给出具体运筹学算法.

文献 [8] 根据代理商物流作业整合的特点, 提出了物流作业的活动整合 (integration) 和批量整合 (consolidation) 的概念. 在文献 [9] 中建立了基于活动整合和批量整合, 选择代理商优化分配物流作业的 0-1 规划模型 (A ICT), 并提出了求解 A ICT 模型的启发式方法和分枝定界法. 但是,

① 收稿日期: 2008-05-08; 修订日期: 2008-10-30.

基金项目: 国家自然科学基金资助项目 (70572028, 70872123).

作者简介: 王 勇 (1957-), 男, 四川内江人, 博士, 教授, 博士生导师. E-mail: wangyongcc@126.com

优化模型 A ICT的数据结构是以矩阵结构为基础的。而实际情况是图状结构,即使是勉强应用矩阵结构也会使模型的矩阵结构特别大,况且许多情况无法刻画。因此,本文在综合考虑代理商选择和线路优化两问题的基础上,建立了基于图状结构的面向第 4方物流的多代理人作业整合优化模型,并根据其特性,设计了求解它的两层邻域搜索算法。

1 基于图状结构的第 4方物流多代理人作业整合优化模型

第 4方物流信息平台,一方面通过网络接受了 M 个作业,另一方面通过网络找到了 S 个物流代理商可供选择。本文基于建立的图状结构下的 0-1 整数规划模型来描述该问题。它考虑如何将每一作业有效地分派给各物流代理商,充分利用作业整合策略,使总费用最低。

设 $G = (V, A)$ 为所考虑的图,其中 V 为节点的集合, A 为边的集合。 G 是所考虑的地理图中物流节点和交通连接的抽象。设有 M 项作业, S 个代理商,作业 m 在第 i 节点与第 j 节点之间的运量只能分给 1 个代理商而不能分给多个代理商。 x_{ijms} 表示第 s 代理商是否承担第 m 项作业在第 i 节点与第 j 节点间的全部运量,取 1 表示承担,取 0 值表示不承担。即

$$x_{ijms} = \begin{cases} 1 & \text{第 } s \text{ 代理商承担第 } m \text{ 项作业在} \\ & \text{节点 } i \text{ 节点 } j \text{ 之间弧的全部运量} \\ 0 & \text{否则} \end{cases}$$

q_{ijm} 为作业 m 在节点 i 与节点 j 之间的运量, $C_{ijs}(q)$ 为代理商 s 在节点 i 与节点 j 之间运送运量为 q 的货物时的单位运价,它取为单调下降表示运量越多单位运价越少,体现物流作业量增加时由于整合作用使成本降低。如果取为阶梯状下降形态体现了货物运输容器的整合结果。 Q_{ijs} 为代理商 s 在节点 i 与节点 j 之间的总运量, U_{ijs} 为代理商 s 在节点 i 与节点 j 之间的运输能力,以 x_{ijms} 为决策变量的优化模型为

$$\min_{x_{ijms}} F = \sum_{s=1}^S \sum_{m=1}^M \sum_{(i,j) \in A} C_{ijs}(Q_{ijs}) q_{ijm} x_{ijms} \quad (1)$$

约束条件为

$$\sum_{f: (i,j) \in A} q_{ijm} x_{ijms} - \sum_{f: (j,i) \in A} q_{ijm} x_{ijms} = b_{im} \quad \forall i \in V, \forall m \in M, \forall s \in S \quad (2)$$

$$\sum_{s=1}^S x_{ijms} = 1 \quad (3)$$

$$Q_{ijs} = \sum_{m=1}^M q_{ijm} x_{ijms} \leq U_{ijs} \quad (4)$$

$$q_{ijm} \geq 0 \quad x_{ijms} = 0 \text{ 或 } 1 \quad (5)$$

下面解释模型的意义。如约束 (4) 所示, Q_{ijs} 是代理商 s 在节点 i 与节点 j 间对各种作业的总运量,是代理商 s 在节点 i 与节点 j 之间的能力约束。而代理商 s 在节点 i 与节点 j 间的成本为 $C(Q_{ijs})$, 随代理商 s 在节点 i 与节点 j 间对各种作业的总运量 Q_{ijs} 的增大而减少,即 $C_{ijs}(Q_{ijs})$ 为 Q_{ijs} 的单调下降函数,表示整合可以节约运输成本的效应。约束 (3) 表示任一作业在任何两节点的运输只能承包给 1 个代理商,由于此点,式 (1) 和 (2) 中的第 s 代理商承担的第 m 作业的运量 $q_{ijm} (\geq 0)$ 与 s 无关。在式 (1) 中,如果 $x_{ijms} = 1$ 代理商 s 将承担第 m 作业在节点 i 与节点 j 之间的运量为 q_{ijm} 的作业,由于单位运量的成本为 $C(Q_{ijs})$, 运量为 q_{ijm} , 故式 (1) 表示完成这 m 个作业的总成本。约束 (2) 是网络流中的流量守恒条件,约束 (2) 中的前一项表示从节点 i 流出作业量之和,后一项表示从节点 i 流入作业量之和,其中 b_{im} 为第 s 个代理商在第 i 节点发出的第 m 个作业的净运量 (发出运量减去进入运量)

$$b_{im} = \begin{cases} > 0 & \text{如果代理商 } s \text{ 在节点 } i \\ & \text{是作业 } m \text{ 的出发点} \\ < 0 & \text{如果代理商 } s \text{ 在节点 } i \text{ 是} \\ & \text{作业 } m \text{ 的收点} \\ = 0 & \text{否则} \end{cases}$$

2 求解多代理人作业整合优化模型的两层邻域搜索算法

该模型是个较为复杂的大型非线性组合优化问题,在现有文献中,未见针对该模型特殊结构的解法。对于大型组合优化问题,最近国际上较为推崇用邻域搜索 (neighborhood search, NS) 算法来处理。邻域搜索算法主要通过产生当前方案的新邻域并按一定方式转移至新邻域状态来完成优化搜索过程。邻域是指从某一解经过微小局部变化和一步移动所能到达的所有其他解的集合。抛射链 (ejection chain, EC) 算法是一种邻域搜索算

法: Pesch^[10] 针对旅行商周游问题 (traveling salesman problem, TSP) 的特点用改进抛射链算法求解该问题. Regro^[11] 利用抛射链算法来解决车辆路线安排问题 (vehicle routing problem, VRP). 文献 [12] 用基于转移 (shift)、交换 (swap) 和环形移动的抛射链算法来求解广义指派问题. 文献 [13] 在抛射链算法的基础上设计了多初始解的并行抛射链算法 (MPEC) 和联合并行抛射链算法 (CPEC). 文献 [14] 提出了基于改进图 (improvement graph) 的大规模邻域搜索 (VLSN) 方法来解决设施定位问题. 通过对以上新搜索方法的研究, 发现这些算法都属于邻域搜索 (NS) 算法, 改进之处在于他们利用抛射链算法以及改进图等方法构造出更有效的邻域结构.

本文尝试用邻域搜索算法求解该问题, 将算法分为两层, 第 1 层指派各物流作业给各代理商, 第 2 层各代理商针对被委派的作业任务进行线路优化. 为了避免将线路优化与代理商选择相分离引起最优性下降, 与文献 [7] 不同, 不是把代理商选择和线路优化作为两个互相分离的阶段来处理, 而是把代理商选择作为第 1 层优化问题, 把线路优化作为第 2 层问题, 即作为第 1 层问题的子优化问题来求解. 第 1 层问题采用类似于文献 [12] 的邻域搜索方法, 而第 2 层采用本文针对该问题构造的特殊的邻域搜索方法. 两层算法的总体框架流程图如图 1 所示.

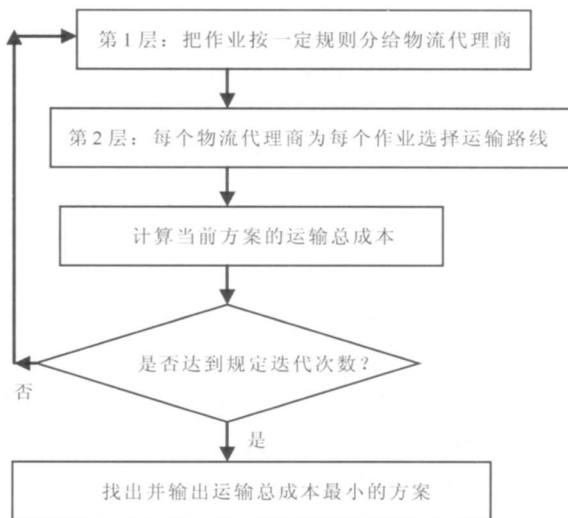


图 1 两层邻域算法总体框架流程图

Fig 1 Flowchart for two-layer neighborhood algorithm

2.1 第 1 层邻域搜索优化算法

用类似于文献 [12] 的转移、交换和环形移动来

求解本文中作业在代理商指派之间的分配问题.

将算法分为两层, 第 1 层的任务是把作业指派给代理商. 所以可以认为第 1 层的解空间是由 {作业 j - 代理商 s } 做主线的, 或者由 {作业 m - 代理商 $s, m = 1, 2, \dots, M, s = 1, 2, \dots, S$ } 所组成. 这样一来, 邻域形成策略是“保持决大多数指派不变, 只变化少量指派”. 下面利用“指派转移”和“指派交换”变化一、二种作业到代理商的指派, 而保持其余指派不变, 形成“指派转移邻域”和“指派交换邻域”的变换. 具体描述如下.

(1) 指派转移 指派转移的方法是从现有代理商中, 随机选择两个代理商 $1(s_1)$ 和 $2(s_2)$, 并从已指派给代理商 $1(s_1)$ 的作业中随机选择一作业 m (Job2), 如果代理商 $2(s_2)$ 的能力能够容纳该作业 m , 则将代理商 $1(s_1)$ 作业 m 转移指派给代理商 $2(s_2)$, 否则重新选择代理商 $1(s_1)$ 的作业. 用 $N_{\text{shift}}(s_1, s_2, m)$ 或用 $N_{\text{shift}}(X)$ 表示指派转移, 其中 X 是当前指派方案.

(2) 指派交换 指派交换就是随机选择两个不同的代理商 $1(s_1)$ 和 $2(s_2)$, 再分别随机选择代理商 $1(s_1)$ 的作业 m_1 和代理商 $2(s_2)$ 的作业 m_2 , 测试进行作业 m_1 和 m_2 对换代理商后, 是否超过代理商 $1(s_1)$ 和代理商 $2(s_2)$ 的物流能力. 如果没有则进行作业指派交换. 用 $N_{\text{swap}}(s_1, s_2, m_1, m_2)$ 或用 $N_{\text{swap}}(X)$ 表示指派交换, 其中 X 是当前指派方案.

(3) 指派环形移动 指派环形移动是由一系列转移移动组成, 是通过同时转移 m_1, m_2, \dots, m_{s_0} (s_0 为选择代理商个数) 作业形成新的解. 即将属于第 1 个代理商的运输作业 m_1 转移给第 2 个代理商, 同时将属于第 2 个代理商的运输作业 m_2 转移给第 3 个代理商, 依次类推, 直到将最后一个代理商的运输作业 m_{s_0} 转移给第 1 个代理商. 用 $N(s_1, s_2, \dots, s_{s_0})$ 或用 $N_{\text{long}}(X)$ 表示指派环形移动, 其中 X 是当前指派方案. 在这种策略中不测试代理商能力是否满足, 该策略用于跳出局部最优解.

用这 3 种策略构造给代理商指派作业的算法时, 首先随机给出初始指派方案 (初始解). 然后应用前 3 个策略, 改进分配方案. 把已比较的方案中相对总成本最小的方案作为当前方案, 如果新

找到的方案的总成本小于当前方案的总成本, 则把新的方案作为当前方案, 通过逐次改进方案以寻求更好的解. 下面给出算法的基本流程.

算法 1(代理商优选算法):

步骤 1 随机分配作业给各代理商作为初始解 X , 循环次数 $T = 1$;

步骤 2 重复应用指派转移策略 N_1 次:

步骤 2.1 用随机选择的方法在 $N_{\text{shift}}(X)$ 中找出满足能力要求的候选指派转移方案 $X_{\text{shift new}}$;

步骤 2.2 调用第 2 阶段程序, 计算候选指派转移方案 $X_{\text{shift new}}$ 总成本;

步骤 2.3 如果 $X_{\text{shift new}}$ 总成本小于原方案的总成本, 置 $X = X_{\text{shift new}}$, 否则转下一步;

步骤 2.4 如果指派转移策略执行次数小于 N_1 , 转步骤 2.1 否则转步骤 3

步骤 3 重复应用指派交换策略 N_2 次:

步骤 3.1 用随机选择的方法在 $N_{\text{swap}}(X)$ 中选出满足能力要求的候选指派交换方案 $X_{\text{swap new}}$;

步骤 3.2 调用第 2 阶段程序, 计算候选指派交换方案 $X_{\text{swap new}}$ 的总成本;

步骤 3.3 如果 $X_{\text{swap new}}$ 总成本小于原方案的总成本, 置 $X = X_{\text{swap new}}$, 否则转下一步;

步骤 3.4 如果指派交换策略执行次数小于 N_2 , 转步骤 3.1 否则转步骤 4

步骤 4 应用指派环形移动策略 N_3 (一般取 $N_3 = 1$) 次:

步骤 4.1 在 $N_{\text{long}}(X)$ 中选出满足要求的环形移动候选方案 X_{long} ;

步骤 4.2 调用第 2 阶段程序, 计算环形移动候选方案 X_{long} 的总成本, 置 $X = X_{\text{long}}$;

步骤 5 循环次数 $T = T + 1$, 如果 T 程序达到指定迭代次数 T_0 , 输出指派方案 X , 终止计算; 否则转步骤 2

在算法 1 开始时, 由于指派比较粗略, 所以用指派转移策略, 到搜索后阶段, 逐步接近局部最优解. 由于指派转移容易破坏代理商能力, 所以该用交换指派这种对代理商能力影响较小的策略, 以求精确搜索. 待基本达到局部最优解后, 用指派环形移动策略以跳出局部最优解, 开始一轮新的搜索. 这种算法看似简单, 但由于其随机搜索性质, 根据文献 [12] 的报道和本文的计算, 有很好的全

局搜索能力.

2.2 第 2 层路线优化算法

经过第 1 层的处理, 第 2 层是针对每个代理商的各作业选择优化运输路线. 因此该模型就转变成多个最小费用多货物流 (multicommodity) 问题. 针对该问题的求解多是将其转换为数学规划, 用数学规划的方法来求解^[15]. 这类方法的一个弱点是没有很好利用图状结构的特点. 在本文中利用邻域搜索算法来求解该问题, 邻域搜索算法用于 TSP 问题较多, 还未见用于此类多货物流问题. 下面针对本文模型的特点, 提出了路合并和路生成替换两种构造多货物流问题邻域的策略.

该问题第 2 阶段的解空间由 {作业 m —路线 r , $m = 1, 2, \dots, M$, $r = 1, 2, \dots, R$ } 所组成. 所以第 2 阶段解的一种邻域是只变化少量作业 (如一个作业) 而保持其余作业不变的运输路线. 下面的路合并策略是把一条路线的作业合并至另一路线, 保持其余路线运输作业不变. 而路生成作业的策略是变化一条运输路线, 保持其余作业路线不变.

2.2.1 路合并策略

整合是物流优化以减少成本的主要手段. 作业运输路线整合是实现物流作业整合的重要手段. 注意到, 在运输方向、货物性质、交货时间、运输方式等因素上显著不同的作业很难整合, 而只有在这些因素上比较接近, 即具有相似性的作业才可能整合. 在进行物流作业整合时, 首先根据物流作业的地理方向特征进行分类, 其次考虑时间要求, 即考虑开始时间、结束时间和对运输时间的要求, 按作业对运输时间的缓急要求可分别采用空运、陆运 (铁路或公路等) 和水运等不同的运输方式, 再次是货物的性质, 如食品和化工品不宜同运. 只有具有相近地理方向、时间要求和货物性质的作业才有希望整合运输. 由于时间要求和货物性质的相似性比较容易判别, 在本文中暂时忽略时间和货物性质的相似性, 或者假设它们是满足的, 重点研究物流作业的地理相似性.

定义 1 若通过某一运输方式 F (空运、铁路或公路、水运) 从 G_A 地可到达 G_B 地, 行驶距离为 $D(A, B)$, 则称 G_A 地与 G_B 在运输方式 F 下的可达距离为 D_{AB} .

在运输方式下的可达距离与 G_A 地和 G_B 地之间的实际距离有区别, 两地间关于某种运输方式的可达距离可能大于其实际距离.

定义 2 给定 ε 是一较小的正数. 运输方式 F , 运输作业 M_A 与 M_B , 它们的运输路线分别为 $R_A : G_{A1}, G_{A2}, \dots, G_{An}$ 和 $R_B : G_{B1}, G_{B2}, \dots, G_{Bm}$, 若 $D(G_{A1}, G_{B1}) \leq \varepsilon$ 和 $D(G_{An}, G_{Bm}) \leq \varepsilon$ 则称作业 A 与 B 对于运输方式 F 在地理上具有完全相似性; 若在运输作业 A 中存在两点 G_{Aj}, G_{Ak} 与作业 B 中两点 G_{Bp}, G_{Bh} 距离 $D(G_{Aj}, G_{Bp}) \leq \varepsilon$ 和 $D(G_{Ak}, G_{Bh}) \leq \varepsilon$ 则称作业 A 与 B 对于运输方式 F 在地理上具有部分相似性.

首先, 判断两作业是否完全相似, 若完全相似, 则将第 2 个作业的路线变成: 从 G_{B1} 走最短路线) 至 G_{A1} , 同第 1 条作业一起从 G_{A1} 至 G_{An} , 再从 G_{An} 走最短路线至 G_{Bm} , 以完成第 2 条作业的运输. 如图 1 所示, 作业 A : 从 G_{A1} 出发, 经 G_{A2}, G_{A3} , 到达 G_{A4} . 作业 B : 从 G_{B1} 出发, 经 G_{B2}, G_{B3}, G_{B4} , 到达 G_{B5} . 但是, 作业 B 可以从 G_{B1} 到 G_{A1} 和作业 A 一起运输, 经 G_{A2}, G_{A3} , 到达 G_{A4} 后, 再单独运到 G_{B5} . 虽然绕道, 但与作业 A 整合运输, 节约的费用可能多于绕道所增加的成本, 从而减少运输成本. 如果不是完全相似, 然后再判断是否部分相似, 如部分相似, 则将第 2 条作业中 $G_{B2}(G_{Bp}), G_{B6}(G_{Bh})$ 之间的路线变成: 从 G_{B2} 走最短路线至 G_{A2} , 同第 1 条作业一起从 G_{A2} 至 G_{A6} , 再从 G_{A6} 走最短路线至 G_{B6} 以完成第 2 条作业的运输, 如图 3 所示.

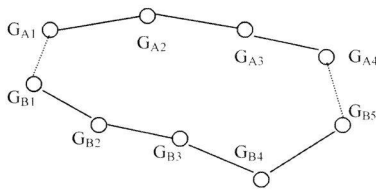


图 2 完全相似作业整合示意图

Fig. 2 chart of complete similar of jobs

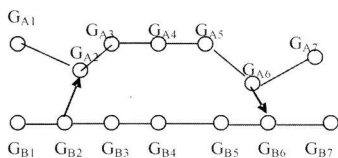


图 3 部分相似作业整合示意图

Fig. 3 chart of segmental similar of jobs

本文用下列算法实现相似路段的寻找与运输路线合并. 判别完全相似性比较容易, 在下列算

法中用步骤 2 实现. 判别部分相似性稍困难, 步骤 3 找出两作业较近的第 1 对节点, 步骤 4 找到两作业较近的第 2 对节点, 步骤 5 完成两对节点之间路线的整合. 下面给出算法的基本流程:

算法 2(路合并算法):

步骤 1 选择两作业路线, 设作业 A 的起点为 G_{A1} , 终点为 G_{An} , 走的路线为 $road(G_{A1}, G_{An})$. 作业 B 的起点为 G_{B1} , 终点为 G_{Bm} , 走的路线为 $road(G_{B1}, G_{Bm})$.

步骤 2 搜索完全相似路径. 随机选中一个作业, 如 A 作为主作业, 然后判断这两个作业起点之间与终点之间的距离是否分别小于某一较小的值 ε 即是否满足 $D(G_{A1}, G_{B1}) \leq \varepsilon$ 和 $D(G_{An}, G_{Bm}) \leq \varepsilon$. 如果满足, 则将作业 B 路线变成: 从 G_{B1} 走最短路线至 G_{A1} , 同主作业一起经 $road(G_{A1}, G_{An})$ 到达 G_{An} , 再从 G_{An} 至 G_{Bm} (走最短路线) 以完成作业 B 的运输; 如果不满足, 转下一步;

步骤 3 搜索部分相似路径的第 1 对节点. 搜索主作业 A 路线和作业 B 路线的所有节点, 找出两作业路线之间距离最近的两节点 G_{Ai} 与 G_{Bj} . 若 G_{Ai}, G_{Bj} 之间的距离小于某一较小的值 ε 即 $D(G_{Ai}, G_{Bj}) \leq \varepsilon$ 则转步骤 4. 否则, 表明该两路径距离较远, 没有相似性, 终止程序;

步骤 4 搜索部分相似路径的第 2 对节点.

步骤 4.1 搜索主作业节点 G_{Ai} 之前所有节点以及第 2 个作业节点 G_{Bj} 之前的所有节点, (前后顺序由作业方向确定), 找出距离最近的两节点 G_{Ak} 和 G_{Bp} , 若 G_{Ak} 和 G_{Bp} 之间的距离小于某一较小的值 ε 即 $D(G_{Ak}, G_{Bp}) \leq \varepsilon$ 转步骤 5. 否则, 转步骤 4.2

步骤 4.2 搜索主作业节点 G_{Ai} 之后所有节点以及第 2 个作业节点 G_{Bj} 之后的所有节点, 找出距离最近的两节点 G_{Ak} 和 G_{Bp} , 若 G_{Ak} 和 G_{Bp} 之间的距离小于某一较小的值 ε 即 $D(G_{Ak}, G_{Bp}) \leq \varepsilon$ 转步骤 5. 否则, 表明两作业没有部分相似性, 终止程序;

步骤 5 路径整合变换.

步骤 5.1 若节点 G_{Bp} 在节点 G_{Bj} 之前, 则将第 2 个作业 G_{Bp} 和 G_{Bj} 之间路线变成: 从 G_{Bp} 至 G_{Ak} (走最短路), 同主作业一起经 G_{Ak} 至 G_{A6} , 再经 G_{A6} 至 G_{Bj} (走最短路线);

步骤 5 2 若节点 G_{Bj} 在节点 G_{Bj} 之后, 则将第 2 条作业节点 G_{Bj} 和 G_{Bj} 之间路线变成: 从 G_{Bj} 至 G_{Ai} (走最短路线)、同主作业一起经 G_{Ai} 至 G_{Ak} , 再从 G_{Ak} 至 G_{Bj} (走最短路线)。

2 2 2 路生成替换策略

与路合并策略不同的是, 该策略是通过变换某一作业的运输路线来改进现有方案。

算法 3 (路生成替换算法):

步骤 1 对于某一作业路线, 首先选择路线上的两点;

步骤 2 利用 k -最短路算法^[16]产生连接两点之间的 K 条最短路;

步骤 3 将作业两点之间路线替换成 K 条最短路中的某一条路线。

2 2 3 总成本函数的计算

下面给出计算具体运输方案总成本算法的基本流程。

算法 4 (总成本函数的计算):

步骤 1 读取方案数组、代理商的运输价格分段函数形式 (表示价格折扣随运输量增大);

步骤 2 根据图结构建立路段流量表;

步骤 3 逐一读入方案数组各路段, 在路段流量表中添加路段流量;

步骤 4 根据各路段流量、运输价格, 返回该代理商的运输成本。

2 2 4 第 2 层路径选择优化算法

第 2 层路径算法的基本思想是先让各作业走最短路线作为初始解, 然后多次利用路径合并策略尽量将作业路线合并, 使货物整合运输以降低成本, 并同时使用路径生成策略使方案多样化。

算法 5 (第 2 层路径选择优化算法):

步骤 1 读取所有代理商的方案数组、图结构;

步骤 2 利用 Floyd 算法, 算出代理商每个运输作业的最短路线, 让每个作业走最短路线以形成初始方案;

步骤 3 调用算法 4 计算初始方案的各代理商运输作业的成本;

步骤 4 $sum = 0$ 对代理商 s 进行循环;

步骤 5 置邻域转换选择迭代次数 $T = 0$ 对 T 进行循环;

步骤 6 策略选择. 选择合并策略或生成替换策略, 若选择合并策略, 转步骤 6 若选择生成替换策略, 转步骤 7;

步骤 7 选择两条未合并的路径, 调用算法 2 通过路合并策略形成尝试方案, 形成复合作业, 转步骤 9;

步骤 8 随机选择一个作业, 调用算法 3 通过生成替换策略, 新生成一条路径, 让选择作业沿新生成路径走, 转步骤 9;

步骤 9 调用算法 4 求出尝试方案中的成本, 若该方案的成本小于当前方案的成本, 取该方案为当前方案, 否则保持当前方案不变;

步骤 10 如果邻域转换迭代次数 w 未达规定次数, $T = T + 1$, 转步骤 6 否则转步骤 11;

步骤 11 令 $sum = sum + \{$ 该代理商 s 的运输成本 $\}$;

步骤 12 若对代理商 s 的循环未结束, 转步骤 3 否则输出该方案, 终止迭代, 返回 sum 。

由于利用路径合并算法和路径生成替换算法每次只改变整个方案的一条路径, 所以仍属于邻域搜索算法。

3 计算实例

为了了解本文提出的两层邻域搜索算法的效果, 最好能把它产生的解同精确解比较. 但如果通过全枚举法求精确最优解, 时间复杂性和空间复杂性都太高, 而且编制程序特别复杂. 为此本文进行了一个非常接近全枚举的基于 k -最短路的枚举方法的计算程序, 把它的解同本文提出的两层邻域搜索算法的解相比较. k -最短路算法可以求出某作业完成的最短路、次短路、第 3 短路、第 4 短路、……、第 $K-1$ 短路和第 K 短路. 由于在最优解中, 虽然可以通过任务整合而走“绕路”降低成本, 但所采用的“绕路”不可能离最短路太远. 因为远离最短路走“绕路”而多走的路所增加的成本会抵消掉作业整合所减少的成本. 所以可以认为基于 k -最短路的枚举方法所求出的最优解是非常接近问题的最优解或就是问题的最优解. 特别是不断增加 K 的值, 如果其解不再减小, 此时, 基于 k -最短路的枚举方法几乎就可以认为是问

题的最优解. 如对每个作业求出 15 条 15-最短路径 (即最短路径、次短路径、...、第 14 短路径、第 15 短路径), 再对每个作业求出 16 条 16-最短路径, ..., 求出 20 条 20-最短路径, 再分别对其代理商分配作业和各种可能路径进行枚举. 至最优解没有变化, 把它作为近似最优解或最优解来进行比较. 在表 1 中, 给出了 6 个运输作业 2 个代理商利用基于 k -最短路径的枚举方法所得到的最优解随 K 变化的过程. 当 $K < 10$ 时, 最优解的目标值随着 K 的增大逐渐减小, 但是当 $K \geq 10$ 时, 其最优解的目标值随着 K 的增大不再变化. 本文后面的例子中, 总是增大 K 足够大至使最优值不再变化持续一段时间, 这些基于 k -最短路径的枚举方法的解可以看成是原问题的近似最优解或最优解.

表 1 基于 k -最短路径的枚举方法的最优值随 K 变化的过程
Table 1 Change procedure of enumeration algorithm based on k -shortest path

K 值	费用最优值
5	76.52
9	74.22
10	73.27
11	73.27
12	73.27
15	73.27
20	73.27

基于 k -最短路径的枚举算法的基本思想是首先利用递归算法找出作业在代理商之间的所有指派方案, 设共生成 T 个方案. 对每个方案 i 进行循环, 对方案中每个固定的代理商 s 和代理商所分配的每个作业 m , 利用 k -最短路径算法生成该作业 K 条最短路径. 然后对代理商分配的作业所生成的所有路径进行枚举, 形成各种运输组合方案. 调用成本函数, 求出各运输组合方案中的成本, 并找出其中成本最小的运输组合方案.

基于 k -最短路径的枚举方法编制程序简单, 求出解答精确. 但是, 由于枚举法的时间复杂性随问题规模增大而迅速 (指数性) 增加, 所以是不实用的. 如表 5 中可以看到对规模较大的问题都因计算时间太长而无法求出, 但本文提出的

算法可以求出. 这构成本文算法存在的一种必要性.

用 Delphi 语言编制了程序, 在 CPU 为赛扬 1.7G, 内存为 384M 电脑上对算法进行了仿真计算. 下面给出 20 个算例的计算情况.

算例 1 是 20 个算例中的一个, 较详细地说明它的情况. 算例 1 的数据由图 4 表 2 及表 3 给出. 图 4 给出了运输网络的连通情况及相邻节点间的距离; 表 2 给出了算例 1 中各运输作业的起点、终点以及运输量; 表 3 给出了算例 1 代理商针对不同的运输量的运输价格以及其最大的运输能力.

表 2 算例 1 的各作业的起点、终点以及运输量
Table 2 Start and destination and quantity of each job

作业	发点	收点	运输量 / 吨
1	1	11	3
2	2	14	4
3	3	15	5
4	4	25	6
5	7	23	6
6	11	20	5
7	13	21	7
8	18	6	8
9	20	16	4
10	26	5	4

表 3 算例 1 的各代理商的运输价格以及其最大的运输能力
Table 3 Price and maximal capability of each agent

代理商	运输量 / 吨	运输价格 / 万元	最大运输能力 / 吨
1	~ 6	0.12	20
	> 6 ~ 10	0.10	
	> 10	0.08	
2	~ 5	0.14	30
	> 5 ~ 12	0.10	
	> 12	0.07	
3	~ 7	0.13	30
	> 7 ~ 14	0.09	
	> 14	0.06	

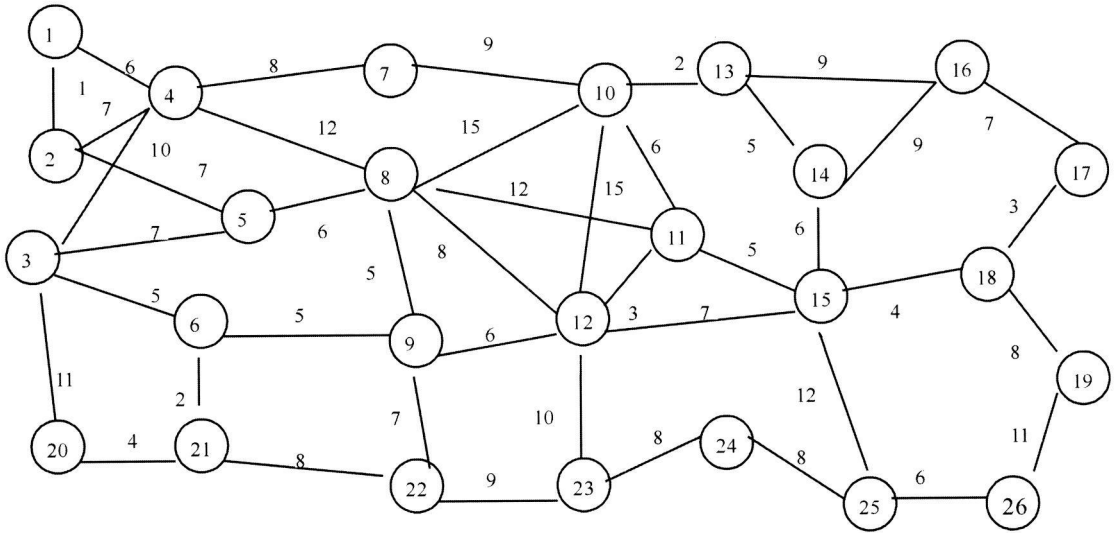


图 4 算例 1 的运输网络图

Fig. 4 Transportation network graph for example 1

表 4 是使用本文提出的两层邻域搜索算法对算例 1 的迭代过程. 首先执行算法 1 的步骤 1 在第 1 轮迭代中根据各代理商的运输能力把作业随机指派给各代理商, 形成的初始指派方案为: 作业 1, 2, 3, 9 由代理商 1 承运、作业 4, 7, 8 由代理商 2 承运、作业 5, 6, 10 由代理商 3 承运. 执行算法 5 的步骤 2 各作业以走最短路作为初始运输路线. 执行算法 5 的步骤 3 计算出代理商 1 的运输成本为 52.72 代理商 2 的运输成本为 52.85 代理商 3 的成本为 53.8 总成本为 159.37 执行算法 5 步骤 6 和步骤 7 针对代理商 1 第 1 轮迭代对 {作业 1, 作业 2} 采用路合并策略. 作业 1 的初始路径: 1-2-5-8-12-11, 作业 2 的初始路径: 2-4-7-10-13-14 选择作业 2 为主作业, 令 $\epsilon = 10$ 根据算法 2 可知作业 1 与作业 2 具有部分相似性, 合并后作业 1 的路径变为: 1-2-4-7-10-11, 代理商 1 的运输成本降低为 51.94 在第 2 轮迭代中, 执行算法 5 的步骤 6 和步骤 7 对 {作业 1, 作业 3} 采用路合并策略, 作业 3 的初始路径: 3-6-9-12-15 选择作业 1 为主作业, 利用算法 2 可得合并后作业 3 的路

径变为: 3-4-7-10-11-12-15, 代理商 1 的运输成本变为 59.18 虽然作业 1, 作业 2 以及作业 3 得到了比较好的整合, 但是成本相对与上一轮迭代增加. 在第 3 轮迭代中, 算法 5 步骤 6 选择作业 2 进行路生成替换策略, 作业 2 路径变为: 2-5-8-12-15-14 (表中未列出). 然后算法 5 循环运用路径合并、生成替换策略, 直到达到迭代次数, 输出代理商 1 的最低运输成本 51.94 同理, 找出代理商 2 代理商 3 的最低运输成本 52.85, 53.8 总成本下降到 158.49, 但成本下降幅度不大, 因此返回到第 1 层改变指派方案, 执行算法 1 并调用算法 5 对各代理商指派的作业运用路径合并、替换策略. 如此反复直到达到迭代次数, 输出算法找到的最低运输成本 137.61.

在表 5 中描述了随机产生了 20 例, 分别用本文的提出的两层邻域搜索算法的解和基于 k -最短路的枚举算法的解进行比较.

由于基于 k -最短路的枚举方法的解可以看成是近似最优解或最优解. 将它与本文算法进行比较, 以计算本算法的误差, 误差为

$$\text{两算法的相对误差} = \frac{|\text{本文两层邻域算法的解} - \text{基于 } k\text{-最短路的枚举方法的解}|}{\text{基于 } k\text{最短路的枚举方法的解}}$$

表 4 算例 1 的两层邻域搜索算法迭代过程

Table 4 Iterative process of two layer neighborhood searching algorithm for example 1

	第一层	第二层	代理商 1				代理商 2				代理商 3			
本文定义的两阶段领域搜索算法的计算结果	第 1 轮迭代	第 1 轮迭代前	作业 1	1-2-5-8-12-11			作业 4	4-8-12-15-25			作业 5	7-10-11-12-23		
		作业 2	2-4-7-10-13-14			作业 7	13-10-11-12-9-6-21			作业 6	11-12-9-6-21-20			
		作业 3	3-6-9-12-15			作业 8	18-15-12-9-6			作业 10	26-25-15-12-8-5			
		作业 9	20-21-6-9-12-11-10-13-16											
												
		局部最优解	作业 1	1-2-4-7-10-11			作业 4	4-8-12-15-25			作业 5	7-10-11-12-23		
			作业 2	2-4-7-10-13-14			作业 7	13-10-11-12-9-6-21			作业 6	11-12-9-6-21-20		
			作业 3	3-6-9-12-15			作业 8	18-15-12-9-6			作业 10	26-25-15-12-8-5		
			作业 9	20-21-6-9-12-11-10-13-16										
本文定义的两阶段领域搜索算法的计算结果	第 2 轮迭代	第 1 轮迭代前	作业 1	1-2-4-3-6-9-12-10-11			作业 4	4-8-12-15-25			作业 5	7-10-11-12-23		
		作业 2	2-4-7-10-13-14			作业 6	11-12-9-6-21-20			作业 10	26-25-15-12-8-5			
		作业 3	3-6-9-12-15			作业 7	13-10-11-12-9-6-21							
		作业 9	20-21-6-9-12-11-10-13-16			作业 8	18-15-12-9-6							
												
		第 2 轮迭代前	作业 1	1-2-4-7-10-11			作业 4	4-8-12-15-25			作业 5	7-10-11-12-23		
			作业 2	2-4-7-10-13-14			作业 6	11-12-9-6-21-20			作业 10	26-25-15-11-12-8-5		
			作业 3	3-4-8-9-12-11-15			作业 7	13-10-11-12-9-6-21						
			作业 9	20-21-6-9-12-11-10-13-16			作业 8	18-15-11-10-7-4-3-6						
												
		局部最优解	作业 1	1-2-5-8-9-12-11			作业 4	4-8-12-15-25			作业 5	7-10-11-12-23		
			作业 2	2-5-8-12-15-14			作业 6	11-12-9-6-21-20			作业 10	26-25-15-11-12-8-5		
			作业 3	3-6-9-12-15			作业 7	13-10-11-12-9-6-21						
			作业 9	20-3-6-9-12-15-14-16			作业 8	18-15-11-12-9-6						
												
本文两阶段领域搜索算法的解			作业 3	3-6-9-12-15			作业 1	1-2-4-7-10-11			作业 6	11-12-9-6-21-20		
			作业 9	20-21-6-9-12-11-10-13-16			作业 2	2-4-7-10-13-14			作业 7	13-10-11-12-9-6-21		
						作业 4	4-7-10-11-12-15-25			作业 8	18-15-11-12-9-6			
						作业 5	7-10-11-12-23			作业 10	26-25-15-11-12-8-5			

由表 5 可见本文建议的两层邻域搜索算法的解与基于 k -最短路的枚举方法的解的平均误差为 1.245%。对规模较小的例子两层邻域算法的解大多数与基于 k -最短路的枚举方法的解相

同。规模稍大的例子本文方法容易达不到最优解,但其解离 k -最短路的枚举方法的解(可视为最优解)也较近。当计算规模再增大时,基于 k -最短路的枚举方法计算时间太长和占用机器内存太

多,发生溢出,得不出最优解.而此时本文建议的两层邻域搜索算法可以计算出最终解,且这些解

比它们的初始解(走最短路的解)有较大改进.本文建议的两层邻域搜索算法的解的改进百分比为

$$\text{两层邻域搜索算法解的改进比} = \frac{\text{各作业走最短路的初始解} - \text{两层邻域算法的最终解}}{\text{各作业走最短路的初始解}}$$

较(各作业走最短路的)初始方案平均下降 10.69%.考虑到本文建议的两层邻域搜索算法的搜索时

间明显小于基于 k -最短路的枚举算法的时间,故本文建议的两层邻域搜索算法有一定的实际意义.

表 5 20例测试例子两层邻域搜索算法的初始解、最终解与基于 k -最短路的枚举算法计算结果的比较

Table 5 Calculating results comparison among the initial solution final solution of two layer neighbourhood algorithm and the k -shortest path enumeration algorithm for 20 examples

例子编号	例子数据特征				两层邻域搜索算法的解				基于 k -最短路枚举算法的解(可视 为精确解)		两种算法费用相对误差 (%) (可视为本 文算法和精 确解的误差)
	作业数	代理商数	节点数	边数	初始解 费用值 /万元	最终费 用值 /万元	计算时 间 /s	相对初 始解下 降 (%)	费用最 优值 /万元	计算 时间 /s	
1	10	3	26	43	159.37	137.61	132.08	13.65	未算出	溢出	—
2	3	1	15	25	37.68	33.16	18.06	12.00	33.16	5.78	0
3	12	3	30	54	195.76	173.92	195	11.16	未算出	溢出	—
4	15	5	30	54	257.84	213.27	280.27	17.29	未算出	溢出	—
5	6	2	18	30	85.2	74.96	86.34	12.02	73.27	244.5	2.28
6	6	3	26	43	76.98	67.23	100.32	12.67	65.04	163.41	3.37
7	13	4	24	44	188.17	167.6	240.36	10.93	未算出	溢出	—
8	7	2	18	30	95.86	82.31	164.11	14.14	82.31	717.25	0
9	11	3	24	44	147.52	136.86	210.33	7.23	未算出	溢出	—
10	8	2	18	30	106.79	99.16	200.07	7.14	未算出	溢出	—
11	9	3	26	43	131.34	118.97	146.12	9.42	未算出	溢出	—
12	18	4	30	54	273.61	233.1	360.27	14.81	未算出	溢出	—
13	10	4	24	44	144.73	138.92	240.51	4.01	未算出	溢出	—
14	4	1	15	25	46.32	44.54	26.12	3.84	44.54	50.16	0
15	15	3	30	54	243.64	216.07	262.84	11.32	未算出	溢出	—
16	5	1	18	30	59.52	56.7	27.51	4.74	55.96	60.57	1.32
17	14	2	26	43	242.44	218.7	199.3	9.79	未算出	溢出	—
18	16	5	30	54	269.54	225.24	300.07	16.44	未算出	溢出	—
19	12	2	24	44	171.64	156.34	200.32	8.91	未算出	溢出	—
20	14	4	26	43	247.02	216.48	280.69	12.36	未算出	溢出	—

4 结束语

本文建立了基于图状结构的面向第 4方物流的多代理人作业整合优化模型.基于物流作业相

似性等性质提出了求解它的两层邻域搜索算法,计算结果表明本文算法具有可行性和有效性.本文未考虑作业在代理商之间转运问题,这有待今后的工作进一步研究.

参考文献:

- [1] 唐加福, Yung Kai-leung 刘士新. 单产品物流网络系统的联合决策模型 [J]. 管理科学学报, 2005, 8(2): 54—60
Tang Jia fu, Yung Kai-leung Liu Shi-xin. Joint decision model for logistic network system with single product [J]. Journal of Management Sciences in China, 2005, 8(2): 54—60 (in Chinese)
- [2] 宁爱兵, 马 良. 竞争决策算法及其车辆路径问题中的应用 [J]. 管理科学学报, 2005, 8(6): 11—18
Ning A-bing Ma Liang. Competitive algorithm and its application to vehicle routing problem [J]. Journal of Management Sciences in China, 2005, 8(6): 11—18 (in Chinese)
- [3] Lau H C, Goh Y G. An intelligent brokering system to support multi-agent web-based 4th party logistics [A]. Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 02). Washington D C., United States, 2002, 154—161.
- [4] 陈建清, 刘文煌, 等. 第四方物流现状及决策支持初探 [J]. 科学学与科学技术管理, 2002, 23(9): 72—74
Chen Jian-qing Liu Wen-huang et al. A survey and study on decision support of 4th party logistics [J]. Science of Science and Management of S & T., 2002, 23(9): 72—74 (in Chinese)
- [5] 陈建清, 刘文煌, 等. 第四方物流中基于多维度的有向图模型及算法 [J]. 工业工程与管理, 2003, 8(3): 45—48
Chen Jian-qing Liu Wen-huang et al. The directed graph model with multi dimensions in the fourth party logistics and its algorithm [J]. Industrial Engineering and Management, 2003, 8(3): 45—48 (in Chinese)
- [6] Vignière G, Amaud K, Michel M. A comparison of heuristics for the discrete cost multi-commodity network optimization problem [J]. Journal of Heuristics, 2003, 9(5): 429—445
- [7] Li Xi-ji, Ying Wei-jun, Liu Wen-huang, et al. The Decision Optimization Model of 4PL [A]. IEEE International Conference on SMC, Washington, 2003, 838—842
- [8] Leung L C, Cheung W, Yee Van Hai. A framework for a logistics e-commerce community network: The Hong Kong air cargo industry [J]. IEEE Transactions on SMC, 2000, 30(4): 446—455
- [9] 王 勇, Lawrence C L, Yee Van Hui. 带时间要求的物流作业集成优化分配模型及求解方法 [J]. 系统工程学报, 2004, 19(2): 154—160
Wang yong Lawrence C L, Yee Van Hui. Assignment optimization method for integration and consolidation of logistics tasks with requirement [J]. Journal of Systems Engineering, 2004, 19(2): 154—160 (in Chinese)
- [10] Pesch E, Glover F. TSP ejection chains [J]. Discrete Applied Mathematics, 1997, 76(1—3): 165—181.
- [11] Rego C. A subpath ejection method for the vehicle routing problem [J]. Management Science, 1998, 44(10): 1447—1559.
- [12] Yagijima Mutsunori, Ibaraki Toshihide, Glover F. An ejection chain approach for the generalized assignment problem [J]. INFORMS Journal on Computing, 2004, 16(2): 133—151.
- [13] Asahiro Yuichi, Ishibashi Masahiro, Yamashita Masafumi. Independent and cooperative parallel search methods for generalized assignment problem [J]. Optimization Methods and Software, 2003, 18(2): 129—141.
- [14] Ahuja R K, Orlin J B, Pallottino S. A multi-exchange heuristic for the single-source capacitated facility location problem [J]. Management Science, 2004, 50(6): 749—760
- [15] Ouhoua A, Mahéy P. A survey of algorithms for convex multi-commodity flow problems [J]. Management Science, 2000, 46(1): 126—147.
- [16] 孙雨耕, 胡华东. 一种应用于路由规划的实用无环 K 路算法 [J]. 计算机工程, 2003, 29(22): 128—130
Sun Yu-geng Hu Hua-dong. A practical loop-less K -path algorithm applied in route planning [J]. Computer Engineering, 2003, 29(22): 128—130 (in Chinese)

Optimization algorithm for multi-agent job integration for fourth-party-oriented logistics

WANG Yong¹, WU Zhi-yong¹, CHEN Xiu-su², WU Jin-qi¹

1. School of Economics and Business Administration of Chongqing University, Chongqing 400044, China

2. School of Mathematics and Statistics of Chongqing Technology and Business University, Chongqing 400067, China

Abstract The integration of logistics job is a main method to reduce the logistics cost. In this paper, we consider the agent and road choice problems comprehensively, and then set up an multi-agent optimization model for job integration based on graph structure and fourth-party-oriented logistics. We propose a two layers neighborhood search algorithm to avoid the separation between agent choices and road optimization. In the first layer, shift move and swap move and loop move are used to assign jobs to agents; in the second layer, the line merging strategy, line generation and replacing strategy are used to form a suitable optimization algorithm for this problem. Twenty examples are generated randomly, to examine the feasibility and validity of the two layers search algorithm, whose results are compared to that of enumerating algorithm based on the k -shortest road.

Key words fourth party logistics; multi-agent job integration; neighborhood search algorithm