

基于扩展状态任务网的制造供应链计划*

苏生⁺, 战德臣, 徐晓飞

(哈尔滨工业大学 计算机科学与技术学院 企业智能计算研究中心, 黑龙江 哈尔滨 150001)

Manufacturing Supply Chain Planning Based on Extended State Task Network

SU Sheng⁺, ZHAN De-Chen, XU Xiao-Fei

(Research Center for Intelligent Computing of Enterprise, School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

+ Corresponding author: Phn: +86-451-86414906, E-mail: susheng@hit.edu.cn, <http://www.hit.edu.cn>

Su S, Zhan DC, Xu XF. Manufacturing supply chain planning based on extended state task network. *Journal of Software*, 2007,18(7):1626–1638. <http://www.jos.org.cn/1000-9825/18/1626.htm>

Abstract: Manufacturing supply chain planning is a key factor of manufacturing supply chain management. Assignment of the production tasks, inventory control and transportation between the plants or the enterprises needs to be considered in manufacturing supply chain planning. Extended state task network (ESTN) is proposed in order to abstractly describe the production tasks with complex product production processes (assembly process, distilling process, process with many kinds of input material and many kinds of output material), the store tasks, and the transportation tasks with different modes (one mode is that transferring only one kind of material and another several kinds of material). In the extended state task network, proportion transform task is used to abstractly describe the production task, the store task and the transportation task that transfers only one kind of material. The combination of virtual proportion transform task and the combination move task are applied to describe the transportation task that transfers several kinds of material. It is easier to encode and operate the solution of the manufacturing supply chain planning if metaheuristic methods are used to solve the problem based on ESTN than on other models. A path relinking algorithm is developed to solve manufacturing supply chain planning model based on ESTN. The algorithm maintains a reference set of solution with good quality in evolutionary process. A list of the new solution (path) is created by inserting properties of a guide solution into an initiate solution. The list is used to update reference set in order to evolve the solution. The solution update method of the reference set with diversification detection and decentralization mutation strategy is proposed in the path relinking algorithm. The results of computations demonstrate that the path relinking algorithm can obtain better solutions than the standard genetic algorithm, the standard Tabu search procedure and the general path relinking algorithm, and prove that the solution update method of the reference set with diversification detection is very important to path relinking. Moreover, the decentralization mutation strategy can enhance the capacity of searching a better solution.

Key words: manufacturing supply chain; planning; extended state task network; path relinking

* Supported by the National High-Tech Research and Development Plan of China under Grant No.2003AA4Z3210 (国家高技术研究发展计划(863)); the Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant No.20030213027 (高等学校博士学科点专项科研基金)

Received 2006-06-09; Accepted 2006-09-27

摘要: 制造供应链计划是制造供应链管理的关键问题,它不仅需要分配生产任务和控制库存,还需要解决不同工厂(企业)间的运输配套问题.为统一描述具有复杂产品生产过程(包括装配型、分解型和多输入多输出型等)的生产任务、存储任务和不同模式(包括单种物料独立运输模式和多种物料组合运输模式)的运输任务,提出了扩展状态任务网(extended state task network,简称 ESTN).扩展状态任务网用比例转化任务统一描述生产任务、存储任务和单种物料独立运输任务,用虚比例转化任务和组合移动任务共同描述多种物料组合运输任务.应用扩展状态任务网,meta 启发式方法在求解制造供应链问题时更容易编码和操作.为求解基于 ESTN 的制造供应链计划模型,提出了具有多样性检测的参考解集更新策略与分散性解变异策略的路径重连算法.路径重连算法维护一个由高质量解(精英解)组成的参考解集,将一个向导精英解的属性逐步引入一个起始精英解而形成的中间解序列(路径),并用此中间解序列更新参考解集以获得进化.计算实例表明,该路径重连算法比标准遗传算法、标准 Tabu 搜索算法以及普通路径重连算法能够获得更好的解,证明了多样性检测对参考解集更新的关键作用以及分散性解变异策略在提高解的质量上的能力.

关键词: 制造供应链;计划;扩展状态任务网;路径重连

中图分类号: TP311 **文献标识码:** A

制造供应链是由原材料供应、制造产品所需的物料转化网络以及产品分销等组成的集成化网络.制造供应链管理的核心是企业内不同业务以及企业间的集成计划和协同运作^[1].这种集成计划模式不仅需要处理复杂的生产任务分配和库存控制,而且还需要考虑不同工厂(企业)间的运输配套问题.其中,运输模式又分两种:单种物料独立运输和多种物料组合运输,这两种运输模式在能力与时间的消耗以及耗用成本上均有较大差别.本文主要研究复杂制造网络环境下的生产与分销集成计划方法.自从 Cohen 开创生产与分销集成计划模式以来,大量研究者进行了相关研究.文献[2]用 Lagrangean 启发式方法研究了多工厂与多客户环境下的产品分配与转移问题.文献[3]提出了能力可扩展条件下的多工厂与多仓库生产计划 MILP(mixed integer linear programming)模型.文献[4]建立了一个单中心工厂与多跨国仓库的集成生产与分销模型.文献[5]研究了用遗传算法求解时间和成本平衡型供应链协同计划.文献[6]提出用基于生成树的遗传算法求解产品生产与分销集成计划问题,并采用 Prufer 数编码生成树解以及使用模糊逻辑控制器实现遗传参数的自动优化调整.文献[7]提出了基于多货物流网络的工厂供应链计划方法,并分析了相应的数学模型,计算复杂性和算法.这些研究的缺点是,它们或者不考虑产品的制造过程,或者仅适用于装配型产品结构,同时都假定运输模式为单种物料独立运输.

针对复杂产品结构,文献[8]将流网络模型应用到有多个制造工厂的生产环境中,并提出了制造网络流的概念,其制造网络的生产过程既可能将多种物料组合转化为另一种物料,也可能将一种物料通过分解获得其他几种物料.在分析最小分销成本流问题的结构与对偶属性后,作者提出了网络单纯形法求解此问题.与文献[8]类似,文献[9]也研究了分解型产品结构下的制造供应链计划问题,在将其转化为一个特殊的最小费用流问题后,提出了改进网络单纯形算法求解此问题.然而,他们没有考虑库存的影响,并同样也都假定运输模式为单种物料独立运输.文献[10]考虑了多种物料组合运输模式,但没有考虑产品的制造过程.

除了基于流网络的形式化模型以外,其他类型的制造供应链计划形式化模型均具有一个共同特点:模型中有区别地表达生产、存储和运输这 3 类任务,并作为 3 种独立变量进行建模,这导致模型求解很困难.特别是在应用 meta 启发式方法求解时难以对众多类型的变量进行统一编码.为此,本文提出统一描述具有复杂产品结构(包括装配型和分解型等)的生产过程、存储过程以及具有不同运输模式(单种物料独立运输和多种物料组合运输)的运输过程的过程模型——扩展状态任务网(extended state task network,简称 ESTN).在将制造供应链问题表达为相应的扩展状态任务网并建立形式化模型后,提出了求解该模型的路径重连(path relinking)算法.

路径重连算法首先被作为 Tabu 搜索的一个组件,由 Glover 在 1994 年提出,用于重新建立 Tabu 搜索过程找到的两个精英解之间的移动轨迹,以实现综合集中性和分散性的搜索过程^[11].之后,路径重连算法被单独抽为一种演化计算方法,其基本思想是,同一个问题的不同精英解之间应该具有一些相似的特征,通过产生精英解之间的解路径以期找到更好的解^[12].最近,路径重连算法被大量应用到求解组合优化等问题上.文献[13-16]将路

径重连算法与贪心随机适应性搜索(greedy randomized adaptive search procedure,简称 GRASP)结合起来,分别用于求解两层图上直线交叉最小化问题、专用虚电路路由问题、矩阵带宽最小化问题以及三维指派问题.文献[17]将路径重连用于求解车辆路由问题.文献[18]将路径重连算法与遗传算法组合用于求解多层仓库选址问题.文献[19]提出了带有弹出链的路径重连算法用于求解广义指派问题.文献[20,21]将路径重连算法与散射搜索(scatter search)算法组合用于系统进化分析和求解有限能力 P -中量问题.文献[22]将路径重连算法与变邻域搜索 VNS(variable neighborhood search)以及 GRASP 算法结合用于求解最大切割问题.与其他演化算法一样,路径重连算法面临的重大问题也是演化解群的多样性问题.为此,本文提出了具有多样性检测的参考解集更新策略与分散性解变异策略的路径重连算法.

1 问题描述

由于在描述具体问题时需要用到状态任务网,故首先介绍状态任务网.在流程制造业(包括间歇过程和连续过程),生产过程通常由配方规定.为准确与简洁地表达配方并便于生产调度建模,Kondili 等人提出了状态任务网 STN(state task network)^[23],并为后来的许多生产调度研究采用^[24].状态任务网是一个有向图,它包含两类节点:状态节点用于表示输入物、半成品和产成品等物质;任务节点用于表示诸如加热、反应和分离等过程操作.状态节点和任务节点分别用圆形和矩形表示,并约定节点之间的有向连线只能发生在状态节点与任务节点之间.一个任务按配方规定的比例消耗输入状态代表的物料,并按配方规定的比例产出输出状态代表的物料,这些比例即为状态节点与任务节点连线上的权重(如果为 1,则省略).由于状态任务网包含状态和任务两类节点,故状态任务网能够清晰地表达装配型、分解型、多输入多产出甚至回流物等复杂产品结构(生产过程结构),为此,采用状态任务网表达生产过程结构.

制造供应链计划问题: M 个制造工厂和 C 个客户组成供需网络 $N=(V,E)$,其中,节点集合 $V=M \cup C$,边集合 $E=\{(u,v,k)|u \in V,v \in V,k \text{ 为自然数,表示节点 } u \text{ 和 } v \text{ 之间的第 } k \text{ 条边}\}$,边 (u,v,k) 上可流动的物料集合为 I_{uvk} ,如果集合 I_{uvk} 的元素个数大于 1,也即边 (u,v,k) 上可流动的物料种类数目大于 1,则表明 I_{uvk} 中的所有不同物料种类可以组合在一起从节点 u 运向节点 v .客户节点 $c(c \in C)$ 的需求物料集合用 I_c 表示.用状态任务网表达产品 i 的工厂级生产过程,其中,工厂节点 m 能够处理的任务集合用 T_m 表示.资源类集合为 R (包括生产、运输和存储资源类),资源类 $r_j(r_j \in R)$ 在周期 p 的可用能力 C_{ip} 用 r_j 在周期 p 的额定的有效作业时间表示.

st_{ij},vt_{ij},sc_{ij} 和 vc_{ij} 分别表示生产类资源 $r_j(r_j \in R)$ 处理任务 $t(t \in T_m, m \in M)$ 耗费的固定准备时间、单位时间、固定准备成本和单位成本.边 (u,v,k) 的运输提前期为 l_{uvk} ,耗费运输类资源 $r_j(r_j \in R)$ 的单位时间、固定成本和单位成本分别为 utt_{uvkj},ftc_{uvkj} 和 utc_{uvkj} .物料 i 在工厂 m 存储耗费存储类资源 $r_j(r_j \in R)$ 的单位库存成本和单位库存时间分别为 ir_{ij} 和 ic_{ij} .在具有 H 个周期的计划区间上,客户 c 在周期 p 对产品 $i(i \in I_c)(1 \leq p \leq H)$ 的需求为 D_{cip} ,其销售单价为 Pr_{cip} .现需决定在任意计划周期 p 上,网络 N 中 $\forall m \in M$ 的生产物料种类、相应数量、节点 $u(u \in M)$ 向节点 $v(v \in N)$ 供给的物料种类和相应数量,使得在满足产品需求与能力约束条件下,企业在计划区间 H 上获得的总利润最大.

图 1 给出了一个由 2 个客户(节点 0 和 1),5 个工厂(节点 2~6)组成的供需网络 N ,工厂级的生产过程由图 2 所示的 STN 表达.其中, $I_{641}=\{2,3\},I_{651}=\{2,3\},I_{421}=\{4\},I_{431}=\{5\},I_{432}=\{4\},I_{521}=\{4\},I_{522}=\{5\},I_{201}=\{6,8\},I_{211}=\{7,8\},I_{301}=\{9\},I_{311}=\{7,9\},I_0=\{6,8,9\},I_1=\{7,8,9\},T_2=\{t_4,t_5,t_6\},T_3=\{t_5,t_7\},T_4=\{t_2,t_3\},T_5=\{t_2,t_3\},T_6=\{t_1\}$.

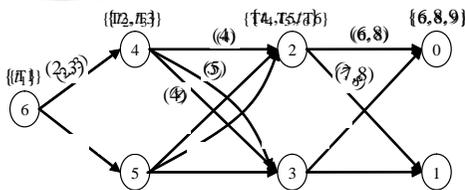


Fig.1 Supply and demand network
图 1 供需网络

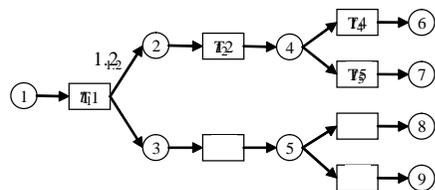


Fig.2 Production processes represented by STN
图 2 STN 表达的生产过程

2 扩展状态任务网及问题模型

2.1 扩展状态任务网

由状态任务网的定义可知,状态任务网不仅能够用于表达生产过程,而且还可以表达存储任务和单种物料运输任务,故用状态任务网统一表达制造供应链过程是适合的.状态任务网用于生产调度的基本条件是状态与任务之间的比例关系(下文定义3中的流权重)必须预知,然而对于多种产品组合运输的运输任务,任务与状态之间的比例关系是未知的,多个输入状态之间以及多个输出状态之间的比例关系也是未知的,故也不能通过组合运输任务的处理量大小来获得输入状态和输出状态的数量.为了建立适应多种物料组合运输任务的制造供应链过程模型,下面提出扩展状态任务网(ESTN).

定义 1. 状态.

状态是对原料、半成品和产成品等物料的抽象表达.在一个扩展状态任务网中,一种实际存在的物料可能与多个状态对应,并用圆形表示状态节点.

定义 2. 任务.

任务是对生产、运输以及存储等活动的抽象表达.任务包括比例转化任务、虚比例转化任务和组合移动任务.比例转化任务指将一种或多种状态转为另外一种或多种状态的活动;虚比例转化任务是为了建立数学模型和求解方便而在扩展状态任务网中虚拟的比例转化任务;组合移动任务是指将多种状态组合在一起,从一个地方移动到另一个地方的活动.在扩展状态任务网中,比例转化任务用实线矩形表示,虚比例转化任务用虚线矩形表示,组合移动任务用八角形表示.

定义 3. 状态任务网 $STN=(S,PT,F,W)$,其中, S 为状态集合, $S \neq \emptyset$; PT 为比例转化任务集合; F 为流集合, $F \subseteq S \times (PT \cup VT) \cup (PT \cup VT) \times S$; W 为流权重集合,设 $f=(r_1,r_2), f \in F$,用 $w_f \in W$ 表示流 f 上的权重,如果 r_1 为状态节点, r_2 为任务节点,则 w_f 或 w_{η_1,r_2} 表示单位任务 r_2 消耗状态 r_1 的比例;反之,如果 r_1 为任务节点, r_2 为状态节点,则 w_f 或 w_{η_1,r_2} 表示单位任务 r_1 产出状态 r_2 的比例.

定义 4. 扩展状态任务网 $ESTN=(S,PT,VT,CT,F,W,C)$,其中, S 为状态集合, $S \neq \emptyset$; PT 为比例转化任务集合; VT 为虚比例转化任务集合; CT 为组合移动任务集合; F 为流集合, $F \subseteq S \times (PT \cup VT) \cup (PT \cup VT) \times S$; W 为流权重集合,设 $f=(r_1,r_2), f \in F$,用 $w_f \in W$ 表示流 f 上的权重,如果 r_1 为状态节点, r_2 为任务节点,则 w_f 或 w_{η_1,r_2} 表示单位任务 r_2 消耗状态 r_1 的比例;反之,如果 r_1 为任务节点, r_2 为状态节点,则 w_f 或 w_{η_1,r_2} 表示单位任务 r_1 产出状态 r_2 的比例; C 为虚比例转化任务与组合移动任务之间的关系集合, $C \subseteq VT \times CT$.

用 n_t 表示任务 t 在某次(周期)的处理量,规定 ESTN 必须满足以下约束:

约束 1. 对 $\text{dom}(F) \cup \text{cod}(F) = S \cup PT \cup VT$,其中, $\text{dom}(F) = \{u \mid \exists v: (u,v) \in F\}$ 为 F 的定义域; $\text{cod}(F) = \{u \mid \exists v: (v,u) \in F\}$ 为 F 的值域,此约束与 Petri 网中约定不能存在孤立元素类似.

约束 2. 对 $\forall v \in CT$,集合 $ST_v = \{u \mid (u,v) \in C\}$ 的元素个数大于等于 2,并称 $\forall u \in ST_v$ 均为任务 v 的从任务,任务 v 为 $\forall u \in ST_v$ 的主任务.

约束 3. 对 $\forall t \in CT$,有 $n_t = \sum_{r \in VT_t} n_r$,表明任意组合移动任务在某次(周期)的处理量等于其所有从任务的处理量之和.

从 STN 和 ESTN 的定义可以看出,ESTN 继承了 STN 定义的状态、任务(ESTN 中为比例转化任务)以及状态与任务之间的流关系与流上的权重,且 ESTN 在 STN 的基础上增加了虚比例转化任务和组合移动任务,其目的是用这两者的组合表达多种物料组合运输任务.此外,状态任务网与 Petri 网在形式上较为类似,但状态任务网相对简洁,仅限于描述静态过程结构,这符合本文的要求,故本文选择扩展状态任务网用于描述制造供应链过程结构.

2.2 问题的ESTN模型

用 $\alpha_{ij}(n_t)$ 和 $\gamma_i(n_t)$ 表示任务 t 对资源类 $r_j(r_j \in R)$ 的能力耗用函数和成本耗用函数, $\beta_i(n_t)$ 表示任务 t 的时间耗用

函数.在不引起歧义的条件下, $\alpha_{ij}(n_i)$, $\beta_i(n_i)$ 和 $\gamma_i(n_i)$ 分别用 α_{ij} , β_i 和 γ_{ij} 代替.基于供需网络 N 和扩展状态任务网 ESTN,使用以下规则可以建立制造供应链计划问题的 ESTN 模型.

规则 1. 对 $\forall t' \in T_m (m \in M)$,将它转化为 ESTN 中一个对等的生产类比例转化任务 t ,将任务 t' 的所有输出状态直接继承为任务 t 的输出状态以及任务 t' 的任意原材料类型输入状态(在 STN 中不存在产出此状态的任务)直接继承为任务 t 的输入状态(状态与任务之间的关系也随之继承).对任务 t' 的任意半成品输入状态 s' (在 STN 中存在产出此状态的任务),在 ESTN 中添加一个新状态 s ,状态 s 为状态 s' 在 ESTN 中的副本,并将状态 s' 与任务 t' 的关系映射为状态 s 与任务 t 的关系. $\alpha_{ij}(n_i)$ 由固定设置耗用能力 st_{ij} 和单位任务耗用能力 vt_{ij} 组成,如式(1)所示, $\beta_i(n_i)=0$,表示通过能力约束,一次生产任务 t 必须在一个周期内完成, $\gamma_{ij}(n_{ij})$ 由固定设置成本 sc_{ij} 和单位任务成本 vc_{ij} 组成,如式(2)所示.

$$\alpha_{ij}(n_i) = \begin{cases} 0, & n_i = 0 \\ st_{ij} + vt_{ij}n_i, & n_i > 0 \end{cases} \quad (1)$$

$$\gamma_{ij}(n_i) = \begin{cases} 0, & n_i = 0 \\ sc_{ij} + vc_{ij}n_i, & n_i > 0 \end{cases} \quad (2)$$

规则 2. 检查网络 N 的任意边,如果边 (u,v,k) 上只流动一种物料 i ,即集合 I_{uvk} 的元素个数等于 1,则添加一个运输类比例转化任务节点,节点的输入状态节点为节点物料对 (u,i) 在 ESTN 中对应的状态节点,输出状态节点为节点物料对 (v,i) 在 ESTN 中对应的状态节点(如果在 ESTN 中不存在相对应的状态节点,则添加一个);如果 $v \in M$,则此任务连接了物料 i 在 STN 中对应的状态映射于 ESTN 中对应的状态及其副本.

如果集合 I_{uvk} 的元素个数大于 1,则添加一个组合移动任务节点和 $|I_{uvk}|$ 个虚比例转化任务节点,符号 \bullet 表示集合 \bullet 的元素个数(下同).对第 $f(1 \leq f \leq |I_{uvk}|)$ 个虚比例转化任务,其输入状态节点为节点物料对 (u,i) 在 ESTN 中对应的状态节点,输出状态节点为节点产品对 (v,i) 在 ESTN 中对应的状态节点,并添加一条指向组合移动任务的有向边.

对 $\forall i \in I_c (c \in C)$ 在 ESTN 中对应的状态,用 D_{sp} 表示客户需求 D_{cip} , Pr_{sp} 表示销售单价 Pr_{cip} ,用集合 $DS = \sum_{c \in C} S_c$ 表示所有客户需求状态.

设边 (u,v,k) 对应的运输任务为 $t \in PT \cup CT$,则其 $\alpha_{ij}(n_i)$ 为单位运输耗用能力 utt_{uvkj} , $\beta_i(n_i)$ 为运输提前期 l_{uvk} , $\gamma_{ij}(n_i)$ 由固定运输成本 ftc_{uvkj} 和单位运输成本 utc_{uvkj} 组成,如式(3)所示.

$$\gamma_{ij}(n_i) = \begin{cases} 0, & n_i = 0 \\ ftc_{ij} + utc_{ij}n_i, & n_i > 0 \end{cases} \quad (3)$$

式(3)中, ftc_{ij} 与 ftc_{uvkj} 对应, utc_{ij} 与 utc_{uvkj} 对应.

规则 3. 对 ESTN 中的任意状态节点 s ,如果既存在输入状态 s 的任务又存在输出状态 s 的任务,则增加一个存储类比例转化任务 t (任务 t 的输入输出状态节点均为 s),则其 $\alpha_{ij}(n_i)$ 为单位库存耗用能力 ir_{ij} , $\beta_i(n_i)=1$, $\gamma_{ij}(n_i)$ 为单位库存成本 ic_{ij} .

图 3 给出了通过规则 1~规则 3 构建的制造供应链计划问题(如图 1 和图 2 所示)的 ESTN 模型.为对比方便,在图 3 所示的 ESTN 中,在每个状态的右下角添加额外的数字表示此状态在图 2 中对应的状态序号.

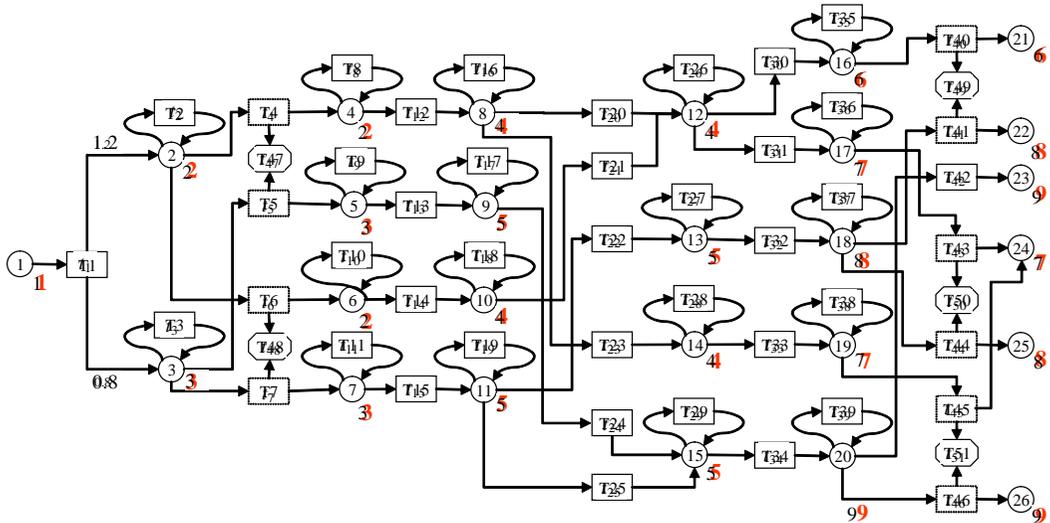


Fig.3 Extended state task network of the manufacturing supply chain process

图 3 制造供应链过程的扩展状态任务网模型

2.3 基于ESTN的形式化模型

用 IT_s 和 OT_s 分别表示输入状态 s 和输出状态 s 的任务集合, T_r 表示消耗资源类 r 的任务集合, 基于 ESTN 的形式化模型如下:

$$\begin{aligned} & \text{Max} \sum_{s \in DS} \sum_{p \in P} Pr_{sp} D_{sp} - \sum_{t \in PT \cup CT} \sum_{r_j \in R} \sum_p \gamma_{ij} (n_{tp}), \\ \text{st.} & \begin{cases} \sum_{t \in OT_s \cap (PT \cup VT)} w_{ts} n_{t,p-\beta_t} = \sum_{t \in IT_s} w_{st} n_{t,p} + D_{sp}, & \forall s \in S, p \in P \\ n_{t,p} = \sum_{r \in VT_r} n_{r,p}, & t \in CT, p \in P \\ \sum_{t \in T_j} \alpha_{ij} (n_{tp}) \leq C_{jp}, & \forall r_j \in R, p \in P \end{cases} \quad (4) \end{aligned}$$

在上述模型中, 目标函数为最大化总利润, 总利润等于销售总收入(第 1 项)减去所有任务的总成本(第 2 项), 式(4)的第 1 个约束为状态平衡约束, 表明所有产出状态 s 的任务 t 在周期 $p-\beta_t$ 的执行量与其产出比例乘积之和(第 1 项)等于所有消耗状态 s 的任务 t 在周期 p 的执行量与其消耗比例乘积之和(第 2 项)加上状态 s 在周期 p 的外部需求量(第 3 项), 式(4)的第 2 个约束表示组合移动任务 t 的处理量等于输入 t 的虚比例转化任务的处理量之和, 式(4)的第 3 个约束为能力平衡约束, 表示 $\forall r_j (r_j \in R)$ 在周期 p 上被消耗的能力不能超过其可用能力 C_{jp} .

3 路径重连算法

本文所研究的制造供应链计划问题是难以求解的, 如果将产品结构简化为装配型产品结构、每种物料由一个确定的资源处理且不考虑运输, 则制造供应链计划问题退化为一个有限能力 MRP(material requirement planning)多层分批问题, 由于有限能力 MRP 多层分批问题是一个 NP-hard 问题, 故多方面复杂化的制造供应链计划问题也一定是一个 NP-hard 问题, 其复杂的状态平衡约束和能力约束使得各变量的取值相互影响, 即使寻找一个可行解也是不容易的, 下面首先描述算法的预处理过程——ESTN 的阶段化, 然后描述求解算法的两个关键点: 问题的表达(编码)策略和路径重连算法.

3.1 ESTN的阶段化

供需网络 N 具有阶段概念,即将产品需求从客户阶段向制造阶段逐阶推进为产品、半成品和原料需求,在将供需网络 N 和生产过程 STN 结合转化为 $ESTN$ 后,为求解方便,需要对 $ESTN$ 进行阶段化.定义阶段为若干状态与输出这些状态的部分任务组成.令当前阶段 l 的状态集合为 LS_l ,任务集合为 LT_l ,已经访问过的状态集合为 VS ,初始化 $l=0,LS_l=DS$ (客户需求状态集合),循环获得以下集合(按顺序)即可获得整个 $ESTN$ 的阶段: $VS=VS \cup LS_l, LT_l=\{t|OS_t \subseteq VS, \exists s \in LS_l \cap OS_t\}, LS_{l+1}=\{s|s \in IS, t \in LT_l\}$. $ESTN$ 的阶段化也可以直接融入问题的 $ESTN$ 模型一节,然而,由于 $ESTN$ 本身并不需要体现阶段性,故将此部分内容单独提出.

3.2 解的表达与分配

为求解制造供应链计划问题,对于任意阶段 l ,从某个周期 p 的状态需求量确定任务处理量是关键.一种方法是检索每个状态,按一定规则给输出此状态的那些任务分配处理量.然而,由于一个任务可能产出多个状态,一个状态又可能由多个任务产出,且多个任务又可能竞争同一资源,故很难找到合适的分配规则.为此,采用另一种方法:首先给阶段 l 上的每个非组合移动任务确定一个分配顺序值 a_{lpt} (如果阶段 l 包含的任务数量为 $|LT_l|$,则 $t \in LT_l$ 的在周期 p 的分配顺序 $a_{lpt} \in \{1, 2, \dots, |LT_l|\}$),然后根据 a_{lpt} 从小到大的顺序确定任务处理量,同时考虑能力约束和所有输出状态的需求约束.

由于一种产品从开始制造到运送至客户需要经过多个计划周期,故任意阶段 l 一定存在一个最晚有效周期 LP_l ,同时,人为估计一个最大持续周期数 CP_l ,表明阶段 l 在最早有效周期 $LP_l - CP_l$ 之前的任何周期上不再存在需求. $LP_l - CP_l$ 的最小值为 0.类似地,资源 r 也存在最晚有效周期 LP'_r 以及最大持续周期数 CP'_r .为此,用所有 $a_{i,j,k}(i=1, \dots, |L|, j=LP_i - CP_i, \dots, LP_i, k=1, \dots, |LT_i|)$ 组成任务分配顺序变量表示解, L 为阶段集合.图 4 给出了解表达模式的示意图.

$$\left| \begin{array}{ccc|ccc} a_{|L|,LP_l-CP_l,1} & \dots & a_{|L|,LP_l,1} & a_{1,LP_1-CP_1,1} & \dots & a_{1,LP_1,1} \\ a_{|L|,LP_l-CP_l,2} & \dots & a_{|L|,LP_l,2} & a_{1,LP_1-CP_1,2} & \dots & a_{1,LP_1,2} \\ \vdots & & \vdots & \vdots & & \vdots \\ a_{|L|,LP_l-CP_l,|LT_l|} & & a_{|L|,LP_l,|LT_l|} & a_{1,LP_1-CP_1,|LT_1|} & & a_{1,LP_1,|LT_1|} \end{array} \right|$$

Fig.4 Encoding model of solution

图 4 解的表达模式示意图

值得注意的是,对 $\forall i \in L$,最早有效周期 $LP_i - CP_i$ 上的任何存储类任务(输入状态与输出状态相同)不再有效,因为 $LP_i - CP_i$ 之前的任何周期不再为周期 $LP_i - CP_i$ 输入任何状态.

- 解的分配

首先对变量 $a_{i,j,k}$ 赋随机值,然后用 $getAmountbyAssign(init(LS_i),init(R))$ 函数确定各任务的处理量,其中, $init(LS_i)$ 为第 1 个阶段(客户阶段)的状态的初始需求, $init(R)$ 为资源的初始可用能力.对 $\forall i \in L, j \in \{LP_i - CP_i, \dots, LP_i\}$,如果可用能力不满足任务处理量需求,则证明解是不可行的,或者舍弃此解,或者实施修补.在路径重连算法中,变量 $a_{i,j,k}$ 的值变化后也用此函数重新确定阶段 i 和相对周期 j 后各任务的处理量.

当在 $getAmountbyAssign$ 中判断资源可用能力能否处理任务的最大量时,如果任务为虚比例转化任务,只要其对应的组合移动任务具有大于 0 的处理量,则不需要扣除固定耗用能力,直接用可用能力除以任务对资源的单位耗用能力即可,反之,则需要扣除固定耗用能力.此外,在对虚比例转化任务分配一定处理量后,将这个处理量加在其对应的组合移动任务的处理量上,对资源可用能力的更新也需要由组合移动任务的处理量是从 0 变为某处理量,还是从大于 0 的处理量增加到另一个更大的处理量来确定是否扣除固定耗用能力.

算法 1. $getAmountbyAssign$ 算法.

```
For i=1 to |L| Do
    For j=LP_i to LP_i-CP_i Do
```

根据 $a_{i,j,k}$ 值的增序排列阶段 i 上的任务集合 LT_i

For $k=1$ to $|LT_i|$ Do

 设 LT_i 的第 k 个任务为 t , 则 $N_{ij}=\min(\max FR_t, \min FS_t)$;

 根据 N_{ij} 减少相应资源在周期 $j-\beta_t$ 的剩余能力和输出状态在周期 j 的保留需求, 增加输入状态在周期 $j-\beta_t$ 的保留需求.

End For

End For

End For

在算法 1 中, $\max FR_t$ 是任务 t 消耗的那些资源在周期 $j-\beta_t$ 的剩余能力可处理任务 t 的最大量, $\min FS_t$ 是为满足任务 t 的输出状态的保留需求对任务 t 提出的最小处理量.

3.3 路径重连算法描述

路径重连算法是一种确定性演化算法, 它维护一个由高质量解(精英解)组成的参考解集, 其核心演化动力为路径重连过程: 从参考解集中任意或按一定规则选择两个精英解分别作为起始解(initiating solution)和向导解(guiding solution), 然后产生一条从起始解延伸到向导解的路径(path)以期获得更优解, 解在这条路径上移动时, 向导解的属性逐步被引入到起始解中以形成一系列新解, 然后利用这些新解更新参考解集. 图 5 给出了路径形成过程示意图.

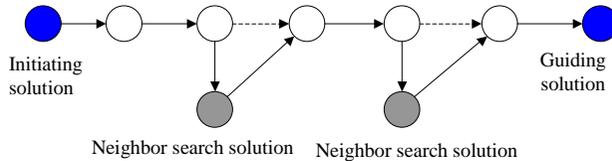


Fig.5 Relinking procedure of path between elite solutions

图 5 精英解之间的路径重连过程

在本文的路径重连中, 路径重连算法用于在能力与需求约束条件下, 通过演化计算来确定最优或近优任务分配顺序值. 算法 2 给出了路径重连算法的主要过程. 本文中选择 $\max Num=10, NumImp=10$.

算法 2. 路径重连算法 PathRelinking.

- (1) 产生多样性解集 $DiverSet$
- (2) 使用 $GetRefSet(DiverSet)$ 策略从多样性解集 $DiverSet$ 获取由 b 个精英解组成的参考解集 $RefSet$
- (3) 根据目标函数值排列参考解集 $RefSet$ 中的解, 使得 x^1 最好, x^b 最差
- (4) 令最好解 $x^*=x^b$, 初始化 $Num=0, \max Num$ 为某正整数, $NumImp$
- (5) $improve=false$
- (6) 使用 $CreateSolPairSet(RefSet)$ 策略从 $RefSet$ 中产生由有序解对集合 $SolPairSet$
- (7) 从集合 $SolPairSet$ 中选择有序解对 (x', x'')
- (8) 应用重连过程 $Relinking(x', x'')$ 产生一个解路径 $Path(x', x'')=x', x(1), x(2), \dots, x(r), x''$
- (9) For $i=1$ to $i < r/NumImp$ Do 对解 $x(NumImp*i)$ 应用改善策略 $Improvement(x)$ End For
- (10) 设新产生的所有解组成解集 $NewSet$, 利用参考解集更新策略 $UpdateRefSet(NewSet)$
- (11) If x^b 比 x^* 好 then $x^b=x^*, improve=true$ End If
- (12) 从集合 $SolPairSet$ 中删除有序解对 (x', x'')
- (13) If $SolPairSet \neq \emptyset$ 返回(8)
- (14) If ($!improve$) then $DiversStrate(RefSet)$ End
- (15) $Num=Num+1$

(16) If Num<maxNum 返回(5) Else 算法停止 End If

3.3.1 参考解集构建策略 GetRefSet

在 PathRelinking 算法中,多样性解集 *DiverSet* 由 h 个随机解组成,由 *DiverSet* 构建参考解集 *RefSet* 不仅需要考虑解的质量,而且需要考虑解集多样性.定义解 u 与解 v 之间的差异度 $D(u,v)$:

$$D(u,v) = \sum_{i \in L} \sum_{j=LP_i}^{LP_i-CP_i} \sum_{k \in LT_i} f(a_{i,j,k}^u, a_{i,j,k}^v),$$

其中,

$$f(a_{i,j,k}^u, a_{i,j,k}^v) = \begin{cases} 0, & \text{if } a_{i,j,k}^u = a_{i,j,k}^v \\ 1, & \text{if } a_{i,j,k}^u \neq a_{i,j,k}^v \end{cases}.$$

GetRefSet 策略按以下步骤获得参考解集:

- (1) 从集合 *DiverSet* 中获得最好解 x^* 加入集合 *RefSet* 中,从 *DiverSet* 中删除解 x^* .
- (2) 对 $\forall x \in \text{DiverSet}$, 计算 $D(x,y)$, 其中 $\forall y \in \text{RefSet}$, 获得解 x 与解集 *RefSet* 中所有解的最小距离 $\min D(x, \text{RefSet})$.
- (3) 将具有最大最小距离的解 $x' (\min D(x', \text{RefSet}) > \min D(x, \text{RefSet}), x \in \text{DiverSet} - x')$ 加入 *RefSet*, 并从 *DiverSet* 中删除解 x' .
- (4) 重复步骤(2)和步骤(3),直至 *RefSet* 的元素个数等于给定元素个数 b .

3.3.2 创建有序解对集合策略 CreateSolPairSet

采用两两组合解的方法从 *RefSet* 创建有序解对集合 *SolPairSet*,确保 *SolPairSet* 中的任意两个有序解对之间至少存在一个不同解,易知,对于元素个数为 b 的 *RefSet*,这种方法将产生 b^2-b 个有序解对.

3.3.3 重连路径方法 Relinking

在从解 x' 向解 x'' 连接时,移动属性为 $a_{i,j,k} (i=1, \dots, |L|, j=|LP_i-CP_i|, \dots, |LP_i|, k=1, \dots, |LT_i|)$, 即每步移动将解 x' 的某个 $a_{i,j,k}^{x'}$ 置为 x'' 的 $a_{i,j,k}^{x''}$, 其中, $a_{i,j,k}^{x'} \neq a_{i,j,k}^{x''}$.

3.3.4 解改善策略 Improvement

采用邻域搜索法实现解的改善,邻域结构为交换邻域 $\text{swap}(e_1, e_2)$, 对 $\forall i=1, \dots, |L|, j=|LP_i-CP_i|, \dots, |LP_i|$, 随机选择 LT_i 上的两个不同位置(任务)分别为 e_1 和 e_2 , 则交换位置 e_1 和 e_2 上的任务分配顺序 a_{i,j,e_1} 和 a_{i,j,e_2} . 这种邻域结构在某个确定周期的 LT_i 上将有 $(|LP_i|(LP_i-1))/2$ 个不同邻域.

3.3.5 参考解集更新策略 UpdateRefSet

这是影响路径重连算法质量的最关键策略,其原因是它对参考解集 *RefSet* 的多样性有极大影响.在 Glover 的基本路径重连算法^[12]中,当一个新解 $x \notin \text{RefSet}$ 且优于 *RefSet* 的最差解 x^b 时,用 x 代替 x^b , 然而,这种只考虑解质量而又不忽略多样性的参考解集更新策略严重削弱了参考解集的多样性,使得某个(些)超级精英解的属性迅速扩散到整个参考解集.令 $x_i (i=1, \dots, |\text{NewSet}|)$ 为 *NewSet* 中的解, $y_j (j=1, \dots, |\text{RefSet}|)$ 为 *RefSet* 中的解, 下面给出综合考虑解质量与解集多样性的 UpdateRefSet 算法, 其中, 常数 *threshold* 为一个差异度阈值, 表示如果一个新解与 *RefSet* 中某个更优解的差异度小于 *threshold*, 即使新解比 *RefSet* 中其他解更好, 也不接收到 *RefSet* 中.

算法 3. 参考解集更新 UpdateRefSet.

For $i=1$ to $|\text{NewSet}|$ Do

 For $j=1$ to $|\text{RefSet}|$ Do

 If $f(x_i) > f(y_j)$ Then

 用 x_i 代替 y_j , 其中, $D(x_i, y_j) \leq D(x_i, y'_j)$, $y'_j \in \text{RefSet} - y_j$ 且 $f(y'_j) < f(x_i)$

 根据目标函数值重排 *RefSet*

 break

 Else

 If $D(x_i, y_j) < \text{threshold}$ Then break; End If;

End If

End For

End For

3.3.6 分散性解变异策略 DiversStrate

从在路径上移动的方法和邻域结构可以看出,对 $\forall a_{i,j,k}$,其赋值范围固定,取值是可记录和统计的.类似于 Tabu 搜索中的基于频率的多样性策略,本文提出的分散性解变异策略 DiversStrate 利用这种统计信息,对 *RefSet* 中的解实施变化,以增强 *RefSet* 中解的多样性.

DiversStrate 方法具体描述:以 *RefSet* 中最差的 $b-1$ 个解为对象,对 $\forall i=1, \dots, |L|, j=|LP_i-CP_i|, \dots, |LP_i|$,从 $|LT_i|$ 个任务中随机选择一个任务 k ,用出现次数最少的值 $least_{i,j,k}$ 代替现有值 $a_{i,j,k}$,如果 $least_{i,j,k}=a_{i,j,k}$,更换另一个任务 k' 重复此过程.

4 计算实例

选择图 1 所示的供需网络和图 2 所示生产过程以及图 3 所示的转化后 ESTN 为计算实例对象,每个工厂具有生产、存储和运输 3 种资源类型,且假定将一个工厂内的所有种类相同的资源作为一个单独的资源处理,不同工厂的同种资源用不同资源序号表示,如用资源 1~资源 3 分别表示工厂 2 的生产、库存和运输资源,用资源 4~资源 6 分别表示工厂 3 的生产、库存和运输资源.从图 1~图 3 可以很容易得知每个任务消耗的资源类型.由于没有可用测试数据集,故随机生成计算实例.任务的固定设置耗用能力为 10~60 之间的随机整数,单位任务耗用能力为 1~4 之间的实数,运输任务处理时间为 1(周期)或 2(周期),固定设置耗用成本为 20~70 之间的随机整数,单位任务耗用成本为 1~4 之间的实数.客户需求周期数量选择 2~8 这 7 种.每个状态在每个需求周期的需求 20~40 之间的随机整数,同一个工厂的某个资源在不同周期上具有相同能力.阶段的最晚有效周期和持续周期数视问题的需求周期数量而定.

对每种客户需求周期数量产生 3 个随机实例,分别标识为 $x-1, x-2$ 和 $x-3$,其中, $x=2\sim 8$.每个实例运行 10 遍.为验证本文提出的路径重连算法的有效性,采用 5 种不同的算法(算法策略)求解每个选定的计算实例,这 5 种算法分别为标准遗传算法 GA(在每个周期相关阶段上实施基于位置的交叉和基因交换变异^[25],初始种群为路径重连算法使用的多样性解集)、标准 Tabu 搜索算法(在每个周期相关阶段上实施交换邻域,初始解为路径重连算法所使用的多样性解集的最好解)、基本路径重连算法、采用本文的参考解集更新策略但不采用分散性解变异策略的路径重连算法以及采用本文的参考解集更新策略和分散性解变异策略的路径重连算法,并分别用 GA, Tabu, PR-G, PR-U 和 PR-D 表示.5 种不同的算法求解各个实例的目标函数值见表 1.其中, Num 列表示客户需求周期数量, No. 表示实例序号, Best 表示 5 种算法所获得的最好解, 每种算法下的 Worst 列和 Best 列分别代表 10 次运行中此算法获得最佳解 S_{best} 和最差解 S_{worst} 与 Best 的差,即 $worst=(S_{worst}-Best)\times 100$, $best=(S_{best}-Best)\times 100$.所有算法采用 Java 语言编程实现,运行环境为一台 PC 机, CPU 为 Pentium(R)4 2.50GHz, 操作系统为 Windows XP.

从表 1 可以看出,在 5 种算法中, PR-D 算法所获得的解质量是最好的,每个实例的最好解均为 PR-D 算法所获得的最优解,其次是 PR-U 算法和 GA 算法. PR-U 算法能够获得多数小问题规模(如需求周期数为 2~4)的最好解.而对于较大规模问题, PR-U 算法获得最好解 Best 的几率越来越小,且其 best 也越来越大,这证明随着问题规模的增加, PR-D 的分散性解变异策略的优势表现得越来越明显,出现这种情况的可能解释是,对于小规模问题, PR-U 算法已经达到了这种启发式模式能够获得的最好解,即使再实施分散性解变异策略,其最好解也不能进一步改善.相对于 PR-U 算法, GA 算法所获得的解质量要差一个数量级左右,且没有获得任何实例的最好解.这充分证明了本文提出的路径重连算法的有效性.求解质量最差的是 PR-G 算法和 Tabu 搜索算法. Tabu 搜索算法在小规模问题上似乎表现得比 PR-G 算法要好,如对实例 2-1~实例 4-3, Tabu 搜索算法获得的最优解均优于 PR-G 算法所获得的最优解,而对于较大规模问题,如对实例 5-1~实例 8-3,除实例 6-1 外, PR-G 比 Tabu 搜索能找到更好解.然而, PR-G 算法还是比 GA 算法要差,表明 PR-G 算法不是一个好算法.事实上, PR-G 算法所具有的较差求解能力正是由于其参考解集的多样性随进化过程急剧下降而导致的.通过观察 PR-G 算法的多样性特征后获

知,由于采用更新参考解集中的最差解,在经过少数重连操作后,PR-G 的参考解集的所有解都变得相同,其原因是,某些超级精英解在进化过程中迅速将自己的属性扩散到参考解集的其他解中,使得进化过程过早收敛,这也证明了本文提出的基于多样性的参考解集更新策略的有效性和重要性。

Table 1 Comparisons of computation results obtained by five algorithms on all instances

表 1 不同算法在计算实例上获得的结果比较

Num	No.	Best	GA		Tabu		PR-G		PR-U		PR-D	
			Worst	Best	Worst	Best	Worst	Best	Worst	Best	Worst	Best
2	2-1	104 05	0.76	0.29	2.58	1.11	3.25	1.89	0.67	0	0	0
	2-2	111 21	0.56	0.12	3.25	1.35	2.85	1.52	0.77	0	0	0
	2-3	115 69	0.62	0.35	2.54	1.89	3.48	1.03	0.98	0	0	0
3	3-1	165 60	2.36	0.59	3.17	1.84	2.35	2.10	0.20	0	0.12	0
	3-2	169 82	2.56	0.62	3.04	1.76	2.21	1.74	0.35	0	0.20	0
	3-3	170 25	2.61	0.74	3.26	1.93	2.59	1.55	0.47	0	0.18	0
4	4-1	227 84	2.50	0.94	4.03	2.16	3.39	1.08	1.05	0	0.46	0
	4-2	210 46	2.12	0.98	3.45	2.66	2.92	1.24	0.55	0.012	0.31	0
	4-3	220 76	2.14	1.02	3.54	2.76	3.67	1.32	0.44	0	0.087	0
5	5-1	282 70	2.42	1.20	3.52	2.85	2.78	1.29	0.59	0.018	0.48	0
	5-1	274 61	2.66	1.51	3.36	2.56	2.62	1.42	0.48	0.033	0.27	0
	5-3	291 04	2.24	1.18	3.57	2.35	3.01	1.63	0.51	0.019	0.35	0
6	6-1	351 35	1.72	1.33	3.71	2.16	2.81	2.37	0.39	0	0.34	0
	6-2	363 43	2.35	1.86	3.12	2.57	3.12	2.43	0.21	0.085	0.097	0
	6-3	357 97	2.10	1.73	3.54	2.33	2.94	1.67	0.45	0.053	0.24	0
7	7-1	438 29	2.34	1.53	3.40	2.55	2.47	1.41	0.21	0.075	0.192	0
	7-2	451 23	2.56	1.76	4.10	3.01	3.02	1.56	0.32	0.11	0.26	0
	7-3	421 59	2.78	2.23	3.44	2.79	2.13	1.27	0.18	0.094	0.15	0
8	8-1	522 50	2.25	0.90	3.64	2.69	3.53	1.80	0.57	0.14	0.42	0
	8-2	546 18	2.98	1.79	3.15	2.62	3.81	2.32	1.03	0.16	0.37	0
	8-3	523 23	2.83	1.64	3.89	2.77	3.21	1.93	0.81	0.099	0.49	0

为进一步了解路径重连算法的进化计算过程,图 6 给出了 PR-D 算法、PR-G 算法和 GA 算法在求解实例 5-1 时所具有的解收敛曲线(100 代进化过程)。对于路径重连算法,用两个解的一次重连作为一个进化代(PR 算法和 GA 算法的一个进化代的耗用时间不一样),且对于 100 代进化过程,PR-D 算法和 PR-U 算法所获得的解一致,故只标注出 PR-D 算法。从图 6 可以看出,PR-D 算法的搜索过程不仅速度快,很快就达到了 GA 算法的相同解质量,而且收敛代数跨度大,即能够改善解的进化过程长,故其获得的解的质量比 GA 算法要好很多。对于 PR-G 算法,虽然其收敛速度也很快,但是在经过少数进化代数后,其解不再改善,出现这种情况也是由于参考解集多样性迅速降低的原因。

图 7 给出了 PR-D 算法和 PR-U 算法在求解实例 5-1 的全过程中参考解集的最优解和最差解的变化趋势(分别用 PR-D(worst),PR-D(best),PR-U(worst)和 PR-U(best)表示)。从图 7 可以看出,PR-D 算法和 PR-U 算法在进化前期趋势相同,其原因是在 PR-D 算法没有调用分散性解变异策略以前,PR-D 算法和 PR-U 算法一样,而由于路径重连算法是一种确定性演化算法,故在初始参考解集相同的条件下,其进化过程相同。在 PR-D 算法中,当重连解不能进一步改善最好解时,调用多样性策略,故 PR-D(best)和 PR-D(worst)进一步改善(进化曲线继续向上波动)。此外,比较 PR-U(best)曲线和 PR-D(worst)曲线可知,在结束进化时,PR-D 算法的参考解集的最差解也接近 PR-U 算法的最好解,从另一角度说明了 PR-D 中分散性解变异策略的有效性。

比较 GA 和 Tabu 搜索算法,PR-D 算法消耗的总时间更长,将 GA 和 Tabu 搜索算法的最大计算代数均设为 100,在实例 5-1 上,GA 和 Tabu 搜索算法的计算时间分别为 17s 和 29s,而 PR-D 算法需要 3 分钟。PR-D 算法需要更多计算时间的原因是起始解与向导解之间的连接过程比较耗时。然而,PR-D 算法获得与 GA 算法相同的解的质量的计算时间却相当短,如图 6 所示,仅仅需要约 10s 的时间 PR-D 算法即可获得与 GA 算法相当的解的质量,这表明,PR-D 算法在计算效率上仍然具有竞争力,但大部分时间用于后续的解改善过程。

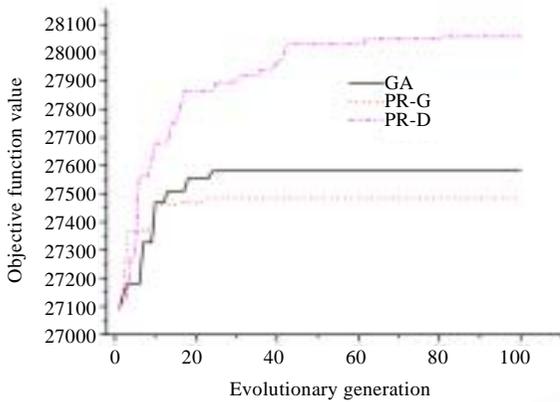


Fig.6 The evolutionary process of PR-D, PR-G and GA

图 6 PR-D,PR-G 和 GA 算法的进化过程

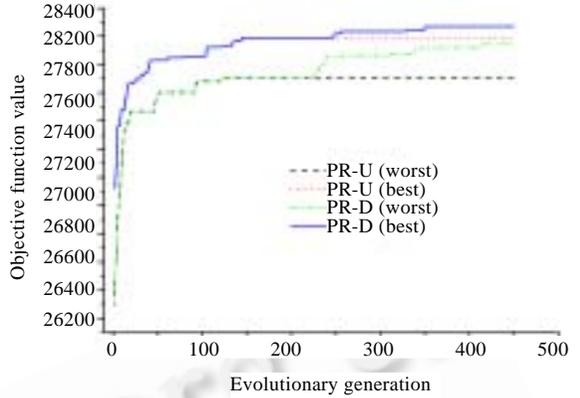


Fig.7 The evolution processes of the best solution and the worst solution in the reference set of PR-D and PR-G

图 7 PR-D 和 PR-U 的参考解集中最好解与最差解的演化过程

5 结 论

本文提出了一种统一描述具有复杂产品结构(包括装配型、分解型和多输入多输出型等)的生产任务、存储任务和不同模式(包括单种物料独立运输模式和多种物料组合运输模式)的运输任务的过程模型——扩展状态任务网,并建立了基于扩展状态任务网的制造供应链计划问题模型.为有效求解此模型,提出了特殊的问题编码机制和具有多样性检测的参考解集更新策略与分散性解变异策略的路径重连算法.通过计算实例证明了路径重连算法的有效性,并分析了不同策略条件下路径重连算法的进化过程.然而,路径重连算法的总计算时间偏长,下一步我们将重点研究其计算效率问题.

此外,还可以通过在不同任务间或者不同周期间微调任务分配量来进一步优化结果,如基于分配数量的邻域搜索,但由于本文的重点在于路径重连算法,故不强调这种微调.

本文也只重点研究供应链集成计划,不涉及企业间的协调和协商问题.基于协商的供应链计划是另一个研究方向.

References:

- [1] Erenguc SS, Simpson NC, Vakharia AJ. Integrated production/distribution planning in supply chains: An invited review. *European Journal of Operational Research*, 1999,115(2):219–236.
- [2] Guinet A. Multi-Site planning: A transshipment problem. *Int'l Journal of Production Economics*, 2001,74(1-3):21–32.
- [3] Jolayemi JK, Olorunniwo FO. A deterministic model for planning production quantities in a multi-plant, multi-warehouse environment with extensible capacities. *Int'l Journal of Production Economics*, 2004,87(2):99–113.
- [4] Özdamar L, Yazgac T. A hierarchical planning approach for a production-distribution system. *Int'l Journal of Production Research*, 1999,37(16):3759–3772.
- [5] Chen HL, Zhang J, Ma DZ. Research on supply chain collaborative planning based on cost and time balance optimization. *Computer Integrated Manufacturing Systems*, 2004,10(12):1518–1522 (in Chinese with English abstract).
- [6] Gen M, Syarif A. Hybrid genetic algorithm for multi-time period production distribution planning. *Computers and Industrial Engineering*, 2005,48(4):799–809.
- [7] Wu SD, Golbasi H. Multi-Item, multi-facility supply chain planning: Models, complexities, and algorithms. *Computational Optimization and Applications*, 2004,28(3):325–356.
- [8] Fang SC, Qi L. Manufacturing network flows: A generalized network flow model for manufacturing process modeling. *Optimization Methods and Software*, 2003,18(2):143–165.

- [9] Mo J, Qi L, Wei Z. A manufacturing supply chain optimization model for distilling process. *Applied Mathematics and Computation*, 2005,171(1):464-485.
- [10] Sakawa M, Nishizaki I, Uemura Y. Fuzzy programming and profit and cost allocation for a production and transportation problem. *European Journal of Operational Research*, 2001,131(4):1-15.
- [11] Glover F. Tabu search for nonlinear and parametric optimization (with links to genetic algorithms). *Discrete Applied Mathematics*, 1994,49(1-3):231-255.
- [12] Glover F, Laguna M, Martí R. Fundamentals of scatter search and path relinking. *Control and Cybernetics*, 2000,29(3):653-684.
- [13] Laguna M, Martí R. GRASP and path relinking for 2-layer straight line crossing minimization. *INFORMS Journal on Computing*, 1999,11(1):44-52.
- [14] Resende MGC, Ribeiro CC. A GRASP with path-relinking for private virtual circuit routing. *Networks*, 2003,41(2):104-114.
- [15] Estefania P, Isaac P, Vicente C, Rafael M. GRASP and path relinking for the matrix bandwidth minimization. *European Journal of Operational Research*, 2004,153(1):200-210.
- [16] Aiex RM, Resende MGC, Pardalos PM. GRASP with path relinking for three-index assignment. *Journal on Computing*, 2005,17(2):224-247.
- [17] Sin CH, Michel G. Path relinking for the vehicle routing problem. *Journal of Heuristics*, 2006,12(1-2):55-72.
- [18] Zhang GQ, Lai KK. Combining path relinking and genetic algorithms for the multiple-level warehouse layout problem. *European Journal of Operational Research*, 2006,169(2):413-425.
- [19] Yagiura M, Ibaraki T, Glover F. A path relinking approach with ejection chains for the generalized assignment problem. *European Journal of Operational Research*, 2006,169(2):548-569.
- [20] Cotta C. Scatter search with path relinking for phylogenetic inference. *European Journal of Operational Research*, 2006,169(2):520-532.
- [21] Diaz JA, Fernandez E. Hybrid scatter search and path relinking for the capacitated p -median problem. *European Journal of Operational Research*, 2006,169(2):570-585.
- [22] Festa P, Pardalos PM, Resende MGC, Ribeiro CC. Randomized heuristics for the max-cut problem. *Optimization Methods and Software*, 2002,7(6):1033-1058.
- [23] Kondili E, Pantilides CC, Sargent RWH. A general algorithm for short-term scheduling of batch operations-I MILP formulation. *Computers and Chemical Engineering*, 1993,17(2):211-227.
- [24] Floudas CA, Lin X. Continuous-Time versus discrete-time approaches for scheduling of chemical processes: A review. *Computers and Chemical Engineering*, 2004,28(11):2109-2129.
- [25] Wang L. Shop Scheduling with Genetic Algorithm. Beijing: Tsinghua University Press, 2003. 40-42 (in Chinese).

附中中文参考文献:

- [5] 陈淮莉,张洁,马登哲.基于成本和时间平衡优化的供应链协同计划研究. *计算机集成制造系统*,2004,10(12):1518-1522.
- [25] 王凌.车间调度与遗传算法.北京:清华大学出版社,2003.40-42.



苏生(1978 -),男,四川南充人,博士生,CCF 学生会会员,主要研究领域为 CIMS,ERP, SCM,生产计划与调度.



徐晓飞(1962 -),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为 CIMS, ERP,数据库,管理与决策信息系统.



战德臣(1965 -),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为 CIMS, ERP,管理与决策信息系统.