

云制造环境下的云企业资源优化配置^①

程 臻^② 战德臣

(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

摘 要 分析了现有云制造环境下云企业资源优化配置方法的局限性,充分考虑了云制造环境下资源的关联性、相似性及先验性特点,从云企业运营费用的角度提出了一种新的云企业资源优化配置方法。该方法首先定义资源优化配置问题的决策变量,建立资源优化配置问题模型,然后建立资源的关联性、相似性及先验性的相应模型并将其应用到目标函数中,最后采用基于实数矩阵编码的遗传算法进行问题的求解。由于采用了适合实数矩阵编码的交叉变异规则和压力标定的遗传策略,该算法能够更快地趋近最优解。试验结果表明,该方法在充分考虑资源的关联性、相似性及先验性的情况下,能够取得较好的资源优化配置结果和较高的算法执行效率,从而验证了其可行性和有效性。

关键词 云制造, 资源优化配置, 资源关联性, 资源相似性, 资源先验性

0 引 言

云制造是服务型制造的一种典型模式,其核心理念是通过服务化、虚拟化将各类制造资源和制造能力形成制造资源云池^[1],在此基础上,云企业^[2]通过资源的选择、组合及调度来完成全生命周期的制造活动^[3]。然而,随着制造资源云池中虚拟资源的数量的增加,云制造的效率会大大降低^[4],因为每一次任务都进行一次重新的资源选择,所以云企业通常会对云服务平台中的已经聚类的资源依据自身任务特性,进一步优化配置自身的资源池,以提高资源的选择、组合与调度的效率。另一方面,与云计算的资源基本可以按照资源的实时状态进行选择、调度^[5,6]不同,云企业对制造资源的选择、调度需要按照资源提供者的制造资源计划进行安排^[7],因而,虽然云企业对资源也是按需使用、按使用付费,但在进行资源优化配置时,通常会与相应的资源提供者签订资源使用合同,甚至预付相应的费用,以保证资源能够按需使用。然而,云企业进行资源优化

配置时,并不是配置越多的资源越好,首先资源配置的越多,其资源选择时检索效率会大大下降。极端情况下,就算云企业配置了云服务平台中的所有资源,结果是云企业资源选择的效率与没有进行优化配置时一样低。其次,由于云企业配置资源时通常要签订合同,因而需要预付相应的资源费用,与制造企业购买生产设备类似,配置太多的资源,会占用云企业大量的资金,导致企业无法进行正常生产。相反,配置太少的资源,会导致资源无法按需使用,也会提高资源的使用成本。因此,人们应研究、优化配置云企业的资源的方法,以保证高效地进行资源选择、调度的同时,尽可能多地完成客户的任务。

对云企业资源的优化配置,现有的研究大都是基于任务属性和资源自身属性进行的^[8-10],而且大量的研究都认为所有任务都能够同时开始执行并且能够完成^[11-14]。然而,在云制造环境中,客户任务可能是持续不断的,在时间上形成了一个任务流,就像制造企业的计划编制有计划周期一样。本研究对任务也采取分阶段批处理,即同时批量地处理一段

① 国家科技支撑计划(2013BAH17F03)和国家自然科学基金(61273038)资助项目。
② 男,1982年生,博士生;研究方向:云制造服务平台,资源虚拟化,虚拟资源聚类,云制造资源配置,云制造资源组合及优化调度;联系人,E-mail: chengzhen_hit@163.com
(收稿日期:2014-12-19)

时间内的任务,并且在满足任务交货需求的情况下,允许任务延迟开始并且中间可以有间断地执行。同时,本研究还重点考虑了资源的关联性,即资源与资源之间、任务与资源之间存在的一些重要的业务过程约束、商务约定等关联关系,主要包括业务关联关系、质量关联关系、任务与资源之间的关联关系等;资源的相似性,即资源与资源之间在某些方面存在的很强的相似性,主要包括功能相似性、非功能相似性;资源的先验性,即通过统计分析、数据挖掘等技术从大量资源使用、组合与调度的记录中得到的有重要应用价值的资源使用经验,包括用户选择资源的偏好、资源组合的惯用方案、资源的惯用组织与调度方案等。考虑关联性可以使问题的求解更加符合问题的实际情况,考虑相似性对于资源配置的优化补充具有一定的指导作用,利用先验性可以快速找到满足需求的服务或资源,从而提高问题的求解效率。本文分析了云制造环境下的任务类型,给出了的问题的假设,在此基础上定义了问题的决策变量,建立了资源优化配置问题模型,同时对资源的上述 3 个特性进行了建模并将其应用到目标函数中,然后采用基于实数编码矩阵的遗传算法来进行求解。该模型的可行性及有效性通过试验得到了验证。

1 问题描述

依据任务划分的子任务和对应资源候选集,以及子任务的执行路径的异同,本文将任务之间的关系划分为 3 类:(1) 不同类型的任务(子任务的执行路径以及对应的候选资源集完全不同);(2) 相同类型的任务(子任务的执行路径以及对应的候选资源集完全相同);(3) 混合类型的任务(子任务的执行路径以及候选资源集部分相同)。

在云制造环境中,一个云企业通常只完成一种特定类型的任务。例如一个云企业只生产 C70E 型敞车,另一个云企业只完成计算机仿真业务等。当然也有一个云企业能够实现不同类型或者混合类型的任务,此时可以将一个云企业划分成多个只完成一种特性类型任务的云企业来处理即可,因此,本文将重点考虑一个云企业只完成一种特定类型任务的

资源优化配置问题。

1.1 问题假设

本文给出如下问题假设:

假设 1:云制造资源池中的所有资源都是特有的,也就是说每个资源都各不相同。若有相同的资源,本文则将其划分为多个资源,只不过这多个资源的功能和质量相似性最高。

假设 2:云企业配置的资源分为两类:一类是需要签订合同的资源,即有维持费用的资源;一类是不需要签订合同的资源,即无维持费用的资源。通常情况下,对于同一个资源而言,没有维持费用的单次使用费用会大于有维持费用的单次使用费用。

假设 3:云企业在资源配置之前,已经按照功能相似性依据子任务划分出了相应的候选资源集。即在资源配置时,主要考虑资源的非功能相似性。

假设 4:云企业在一个周期内的任务服从泊松分布,且需要完成所有任务,但允许有任务延期,同时会有相应的延期罚款。

假设 5:云企业所配置的资源,除了具备独占性外,还可以在一个周期内在所有任务中分时共享。

1.2 决策变量

基于上述假设,云企业在资源配置时只需要在相应候选资源集中选择需要维持哪些资源,哪些资源不需要维持但可能会使用,以及哪些资源根本不需要维持也不会使用。因此我们给出两个决策变量 $\mathbf{X}, \mathbf{Y}$,它们用下式表示:

$$\mathbf{X} = \begin{bmatrix} x_{11} & \cdots & x_{1J} \\ \cdots & x_{ij} & \cdots \\ x_{I1} & \cdots & x_{IJ} \end{bmatrix}, \mathbf{Y} = \begin{bmatrix} y_{11} & \cdots & y_{1J} \\ \cdots & y_{ij} & \cdots \\ y_{I1} & \cdots & y_{IJ} \end{bmatrix} \quad (1)$$

其中 $x_{ij}, y_{ij} \in \{0, 1\}$ 。

上述两个决策变量均为 I 行 J 列的 01 矩阵, J 列表示任务的 J 个子任务或候选资源集, I 行表示每个子任务所对应的候选资源集中均有 I 个候选资源。

其中决策变量 $\mathbf{X}$ 为云企业需维持的资源矩阵,若 $x_{ij} = 1$,则表示需维持该资源,否则不需要维持。决策变量 $\mathbf{Y}$ 为云企业不需维持但可能使用的资源矩阵,若 $y_{ij} = 1$,则表示不需维持但可能使用的该资源。根据假设 2,决策变量应满足如下约束:

$$x_{ij} + y_{ij} \leq 1 \quad (2)$$

上述约束表示某一个资源只有三种可能性:被维持;不被维持,但可能被使用;既不需要维持也不会被使用。

1.3 问题模型

基于上述假设和决策变量,问题描述如下:

当一个云企业预期的任务为相同类型的任务,且任务服从 λ 的泊松分布时,每个任务均划分成 J 个子任务,每个子任务均由一个相应的候选资源集 VRS_j 中的一个资源来执行,每个候选资源集 VRS_j 中有 I 个资源 $\{VR_j^1, VR_j^2, \dots, VR_j^I\}$, 每个资源 VR_j^i 的维持费用为 C_{ji}^1 , 在有维持费用的情况下,一次使用费用为 C_{ji}^2 , 在没有维持费用的情况下,一次使用费用为 C_{ji}^3 。云企业的预期资源维持费用最大为 c_{cf} , 那么在允许任务延期的情况下,云企业资源配置的目标就是使得云企业在一段期间内,完成任务的同时使得云企业的总成本最低,以达到最好的生产效益。因此其目标函数如下:

$$\begin{aligned} \text{Min}(\sum_{j=1}^J \sum_{i=1}^I (x_{ij} \cdot C_{ji}^1 + x_{ij} \cdot n(x_{ij}, y_{ij}) \cdot C_{ji}^2 \\ + y_{ij} \cdot m(x_{ij}, y_{ij}) \cdot C_{ji}^3) \\ + \sum_{s \in \Lambda} \omega_s \cdot \text{Max}(t_s(x_{ij}, y_{ij}) - t_s, 0)) \quad (3) \end{aligned}$$

其中: $n(x_{ij}, y_{ij})$ 为云企业资源优化配置下,有维持费用的资源使用次数; $m(x_{ij}, y_{ij})$ 为云企业资源优化配置下,无维持费用的资源使用次数; ω_s 为任务 s 延期的罚款权重; $t_s(x_{ij}, y_{ij})$ 为云企业资源优化配置下任务 s 的交货时间; t_s 为任务 s 预订交货时间。同时,资源维持费用满足约束条件如下:

$$x_{ij} \cdot C_{ji}^1 \leq c_{cf} \quad (4)$$

为了满足服从 λ 的泊松分布的所有任务,目标函数中云企业的总成本包括资源的维持成本和使用成本,以及在资源优化配置的情况下可能出现的任务延期成本。延期成本决定于延期的罚款权重和延期时间,若任务提前完成,则没有延期成本,也没有奖励。

从目标函数中看出,调整企业的资源配置变量 $\mathbf{X}$ 和 $\mathbf{Y}$, 会在改变资源维持成本的同时改变资源的调度方案,导致任务的使用成本及延期成本发生变化。那么在不同资源配置环境下,各任务的资源配

置可依据文献[15]的资源组合及调度方法进行模拟仿真,得到任务的延期时间、延期成本等。至此,云企业的资源配置对维持成本及延期成本以及目标函数的影响可以最终确定。

1.4 模型分析

在目标函数中,本文还将重点考虑资源的关联性以及非功能相似性来进一步优化云企业的资源配置。

(1) 资源的关联性

前文中提到考虑资源的关联性能使得问题更加符合实际的情况。一般来说,资源之间的关联性,可直接由资源提供商给出,也可以通过对资源的历史使用记录采用聚类的方法进行分类,然后基于贝叶斯推理、统计分析的理论及方法来提取,并对关联关系进行分类。以下分别对资源间的业务关联和质量关联进行建模。

资源间的业务关联关系建模如下:

$$\text{RelB}(VR_k, VR_l) = \begin{cases} 1, & VR_k \text{ 与 } VR_l \text{ 业务关联} \\ 0, & VR_k \text{ 与 } VR_l \text{ 业务不关联} \end{cases} \quad (5)$$

当 $\text{RelB}(VR_k, VR_l) = 1$ 时,表示使用 VR_k 必须使用 VR_l 。这就使得在资源配置时,若满足业务关系的资源对中某一资源被维持,那么另一个资源一定会被维持。

资源间的质量关联关系建模如下:

$$\text{RelQ}(VR_k, VR_l) = \begin{cases} 1, & VR_k \text{ 与 } VR_l \text{ 质量关联} \\ 0, & VR_k \text{ 与 } VR_l \text{ 质量不关联} \end{cases} \quad (6)$$

$\text{RelQ}(VR_k, VR_l) = 1$, 表示使用 VR_k 对使用 VR_l 的服务质量有影响。那么对于某一资源 VR_k 的第 t 个质量属性 $q_{VR_k}^t$ 可以采用文献[15]中如下的模型:

$$q_{VR_k}^t = \{ \text{default: Value}_0; VR_1: \text{Value}_1; VR_2: \text{Value}_2; \dots; VR_z: \text{Value}_z; \} \quad (7)$$

式中 Value_0 为默认值,其余 z 个值为与 VR_k 质量关联的资源使得 VR_k 的第 t 个质量属性值。质量属性又分为两类:正向属性和反向属性。其中正向属性表示值越高越好,比如资源可靠性、连接成功率等。反向属性表示值越低越好,如时间、成本、费用等。因此对于属性我们可以给出如下模型:

$$q'_{VR_k} = \begin{cases} \text{Min}\{Value_0, \dots, Value_z\}, & q'_{VR_k} \text{ 为反向属性} \\ \text{Max}\{Value_0, \dots, Value_z\}, & q'_{VR_k} \text{ 为正向属性} \end{cases} \quad (8)$$

显然,目标函数中的 C_{ji}^1 、 C_{ji}^2 、 C_{ji}^3 均属于反向属性,因此可以给出维持费用及使用费用的模型如下:

$$\begin{aligned} C_{ji}^1 &= \text{Min}\{\text{DefaultValue}, \dots, Value_z\} \\ C_{ji}^2 &= \text{Min}\{\text{DefaultValue}, \dots, Value_z\} \\ C_{ji}^3 &= \text{Min}\{\text{DefaultValue}, \dots, Value_z\} \end{aligned} \quad (9)$$

在目标函数中引入该模型即可得到考虑资源关联性条件下的优化的资源配置。

(2)资源的相似性

根据假设 3,本文在云企业资源配置上主要考虑资源的非功能相似性,即质量相似性。质量相似性可以基于语义相似度、数值之间的距离度量等方法给出资源之间的质量相似度模型,根据模型找出具有相似性的服务。本文给出如下质量相似度模型:

$$\text{SimQoS}(VR_k, VR_l) = \sum_t \frac{|q'_{VR_k} - q'_{VR_l}|}{\text{Max}(q'_{VR_k}, q'_{VR_l})} \quad (10)$$

式中, q'_{VR_k} 表示资源 VR_k 第 t 个质量属性, q'_{VR_l} 表示资源 VR_l 第 t 个质量属性, $\text{Max}(q'_{VR_k}, q'_{VR_l})$ 表示资源 VR_k 和 VR_l 中第 t 个质量属性的最大值。设 δ 为功能相似性阈值, $\text{SimQoS}(VR_k, VR_l) \geq \delta$ 表明资源 VR_k 和 VR_l 之间存在服务质量相似性。因此,资源配置变量 X, Y 可以在所有服务质量相似的候选资源中选择出目标函数中 $C_{ji}^1, C_{ji}^2, C_{ji}^3$ 最小费用的资源,同时还要考虑资源的关联性,最后能够收敛到既考虑资源关联性又考虑资源相似性的优化的资源配置。

至此,云制造环境下的云企业资源优化配置模型为式(3) subject to 式(1), (2), (4), (9)。其中式(3)为目标函数,即一段时间内企业运营的总成本最低;式(1), (2)为资源配置变量,描述了资源的配置状态;式(4)表示资源要满足维持费用约束;式(9)表示考虑了关联性及相似性后,资源的三种费用取值。

显然,该问题属于 NP-Hard 问题。

2 虚拟资源优化配置

本文采用考虑资源先验性的遗传算法对云制造

环境下的云企业资源优化配置问题进行求解。遗传算法借鉴了自然界中生物“优胜劣汰,适者生存”的规律,通过对问题的解空间进行编码和对决策变量的运算,在整个解空间有导向性地逼近最优解,为复杂约束条件下求解资源配置问题提供了方法。

2.1 染色体编码

问题模型中存在两个决策变量,为了简化问题的求解,本文给出如下实数矩阵:

$$A = \begin{bmatrix} a_{11} & \dots & a_{1J} \\ \dots & a_{ij} & \dots \\ a_{I1} & \dots & a_{IJ} \end{bmatrix} \quad (11)$$

矩阵 A 有 J 列,表示云企业所需求的 J 个候选资源集;矩阵 A 有 I 行,表示每个候选集中均有 I 个候选资源。当然,并不一定所有的候选资源集中资源数量都一样, I 的取值为所有候选资源集中资源数量最大的数。

矩阵中 a_{ij} 的取值为 $\{0, 1, 2\}$ 集合中的数。该编码的取值与决策变量之间的关系如下:

- (1) 若 $a_{ij} = 0$, 则 $x_{ij} = y_{ij} = 0$, 表示资源既不需要维持,在调度过程中也不会被使用。
- (2) 若 $a_{ij} = 1$, 则 $x_{ij} = 0, y_{ij} = 1$, 表示资源不需要维持,但在调度过程中可能会被使用。
- (3) 若 $a_{ij} = 2$, 则 $x_{ij} = 1, y_{ij} = 0$, 表示资源需要维持,在调度过程中也可能被使用。

可以看出上述编码的方式,是天然满足约束式(1)和式(2),同时对于候选集中资源数量小于所有候选集中资源最大数量时,矩阵中的变量均用 0 来补充。

2.2 初始种群产生

传统初始种群一般是随机产生的。为了加快算法的收敛速度,本文将资源的先验性加入到初始种群的产生。

针对云企业的任务类型,收集其资源选择及组合的历史运行记录,通过数据挖掘、贝叶斯推理等方法,找出最具有相关性的历史数据,然后基于统计分析的方法计算先验性阈值,满足先验性阈值的资源即为该云企业的先验性资源。因此,本文建立如下的资源先验性模型:

$$p_{ij} = \frac{used_{ij}}{req_{ij}} \quad (12)$$

其中 req_{ij} 表示任务请求资源 VR_{ij} 的总次数, $used_{ij}$ 表示某一资源被选择的次数, 设 η 为先验性阈值, $p_{ij} \geq \eta$ 表明该资源 VR_{ij} 是关于任务的先验性资源。

因此, 为了保证个体的随机性, 避免陷入局部最优, 同时又加快算法的收敛速度, 初始种群中的每个个体均是按列随机产生的, 每列的产生方式如下:

(1) $J = 1$ 时, 用以下的方法生成矩阵的第一列; 首先将所有满足先验性的资源的值设为 2, 其余的值则用一个 $\{0, 1, 2\}$ 的随机序列作为矩阵的第一列。此时需注意, 若候选集中资源数量 j_i 小于 I 时, 则第一列中最后的 $I - j_i$ 个值均为 0。

(2) $J > 1$ 时, 每行的值按照以下方式随机生成: 若满足资源先验性, 则直接取值为 2, 否则考虑资源关联性, 即需满足式(5)。若相关联的资源取值为 2, 则直接取值为 2; 若相关联的资源取值为 1, 则在 $\{1, 2\}$ 中随机生成; 若相关联的资源取值为 0, 则在 $\{0, 1, 2\}$ 中随机生成。此处相关联的资源为已经生成的所有的 $J - 1$ 列资源。

(3) 考虑资源维持费用约束, 即需满足约束式(4), 若不满足, 则需要用约束性满足修复算法(算法 1)进行修复。

算法 1

```

1) For 1 to J //按列循环
2) For 1 to I //按行检查
3) 若  $a_{ij} = 0 \parallel 1$ , 则直接跳到第 2 步;
4) 若  $a_{ij} = 2$ , 若  $p_{ij} \geq \eta$ , 则跳到第 2 步;
5) 否则, 若相关联资源的取值为 2, 则跳到第 2 步;
   若相关联资源的取值为 1, 则直接设置  $a_{ij} = 1$ ; 若满足约束式(4), 则直接跳出修复程序;
   若相关联资源的取值为 0, 则  $a_{ij}$  在  $\{0, 1\}$  中随机生成; 若满足约束式(4), 则直接跳出修复程序;
6) For End
7) For End

```

(4) 若修复后依然不满足约束式(4), 则采用不考虑先验性的随机生成方法产生初始种群。

2.3 交叉与变异

因为矩阵采用实数编码, 故传统的布尔值交叉变异方法不足以有效地解决本文的编码交叉和变异, 因此本文设计了以下几种交叉变异算子:

(1) 行交叉

随机生成一个 I 维的 0-1 列向量 V , 若 $V_i = 1$, 则将父项的两个可行解的第 i 行进行交换, 否则不变, 最终生成新的两个子项。因为进行行交叉之后, 可能会出现不满足式(5)的资源关联性约束, 破坏了约束性规定, 因此, 需对生成的新的子项用约束性满足修复算法(算法 2)进行修复。

算法 2

```

1) For 1 to J //按列循环
2) For 1 to I //按行检查
3) 若  $a_{ij} = 2$ , 则直接跳到第 2 步;
4) 若  $a_{ij} = 1$ , 若相关联资源的取值为 2, 则直接设置  $a_{ij} = 2$ ; 否则不变;
5) 若  $a_{ij} = 0$ , 若相关联资源的取值为 2, 则直接设置  $a_{ij} = 2$ ; 若相关联资源的取值为 1, 则  $a_{ij}$  在  $\{1, 2\}$  中随机生成; 否则不变;
6) For End
7) For End

```

在上述关联性约束修复完成后, 还需要考虑资源维持费用约束, 即约束式(4), 若不满足, 则需使用算法 1 进行修复, 若修复后依然不满足, 则直接舍弃新生成的子项。

(2) 列交叉

与行交叉类似, 随机生成一个 J 维的 0-1 行向量 V , 若 $V_j = 1$, 则将父项的两个可行解的第 j 列进行交换, 否则不变, 最终生成新的两个子项。同样可能会出现不满足式(5)和式(4)的情况, 则同样需要使用算法 1 和算法 2 进行修复。

(3) 行变异

因为矩阵采用的是实数编码, 所以传统的布尔值的翻转变异无法有效地解决本文的变异规则, 因此本文设计了如下的变异规则: 对于非零的值进行如下变异:

$$a_{ij} = 3 - a_{ij} \quad (13)$$

对于零值则在 $\{0,1,2\}$ 中随机生成。

因此,对于行变异,首先随机生成一个 I 维的 $0-1$ 列向量 V ,若 $V_i = 1$,则将父项第 i 行所有值按照上述变异规则进行变异,同时注意到每一列的最后的 $I-j_i$ 个值始终为 0 ,不受变异规则的影响,最终生成新的子项。变异之后,可能会出现不满足式 (5) 和式 (4) 的情况,则需要使用算法 1 和算法 2 进行修复。

(4) 列变异

与行变异类似,随机生成一个 J 维的 $0-1$ 行向量 V ,若 $V_j = 1$,则将父项的第 j 列按照上述变异规则进行变异,同时注意到第 j 列的最后的 $I-j_i$ 个值始终为 0 ,最终生成新子项。同样地,需要使用算法 1 和算法 2 进行修复。

2.4 适应值函数及遗传策略

使用上述算法时,所有不满足约束条件的解均在过程中被遗弃,这就使得本文算法所得出的解均是可行解并且都满足了所有的约束条件,因此本文直接将目标函数作为适应值函数,即适应值函数如下:

$$F = \sum_{j=1}^J \sum_{i=1}^I (x_{ij} \cdot C_{ij}^1 + x_{ij} \cdot n(x_{ij}, y_{ij}) \cdot C_{ij}^2 + y_{ij} \cdot m(x_{ij}, y_{ij}) \cdot C_{ij}^3) + \sum_{s \in \lambda} \omega_s \cdot \text{Max}(t_s(x_{ij}, y_{ij}) - t_s, 0) \quad (14)$$

本文的种群遗传策略采用传统的正比选择策略,但是对适应度函数增加了压力标定,从而能够使得算法更快地趋近最优解。

对种群为 $Size$ 中的某个个体 s ,若其适应值为 F_s ,则其选择的概率为

$$\rho_s = \frac{F_s - \chi \cdot \text{Min}(F)}{\sum_{s=1}^{Size} (F_s - \chi \cdot \text{Min}(F))} \quad (15)$$

其中 χ 为压力系数,取值范围为 $[0,1]$,取值越大,则选择压力越大。

3 实验结果及分析

本研究以中国北车生产加工的 C70E 型敞车的加工过程为例进行了试验。生产一台 C70E 型敞车通常需要车轮加工、钩锁加工、车体加工、整车组装、喷涂、试验交付等 6 个主要过程,对其分析中分别考虑和综合考虑资源的关联性、相似性、先验性。每个加工过程的候选资源集 $VRS(VirtualResourceSet)$ 都已经产生,为简化算法的运行,假设每个候选集中均有 100 个资源(若每个候选集中的资源数量不一致,则也可以按照本文算法进行处理),如图 1 所示。

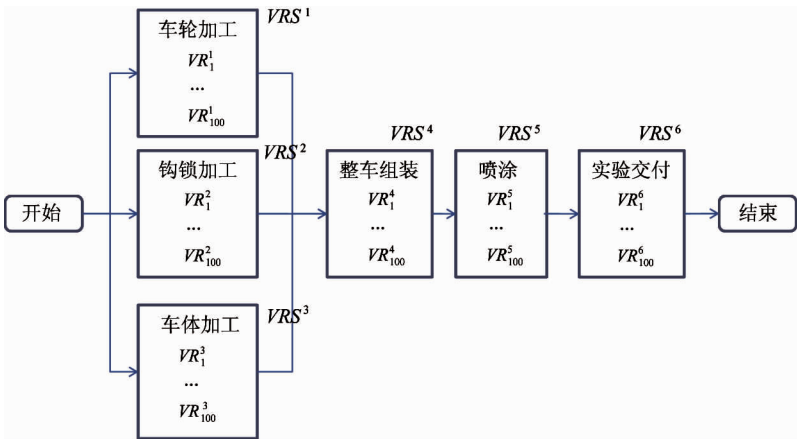


图 1 C70E 型敞车加工过程及候选资源集

在算法分析之前,本研究将随机地生成每个资源的加工周期、维持费用、一次使用费用等信息,同时给定相应的资源关联关系、相似关系以及先验性

条件。同时考虑任务的随机变动因素,以泊松分布模拟任务量,以正态分布模拟任务的交货周期及价格信息。在资源优化配置的情况下,对任务的调度

过程进行仿真试验,计算目标函数,对配置效果进行评价。以下分别按照资源的关联性、相似性、先验性以及综合情况进行仿真试验分析。

3.1 考虑资源关联性的试验及结果分析

在不考虑资源的相似性和先验性的情况下,只考虑资源关联性对最终目标函数的优化结果及算法效率来进行分析。对以下3种情况分别进行了试验:(1)不考虑资源关联性 E1;(2)只考虑资源业务关联性 E2;(3)既考虑业务关联性又考虑质量关联性 E3。

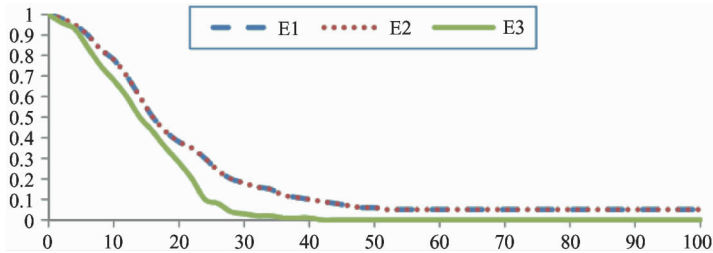


图2 考虑资源关联性的试验结果

图2将总成本按以下公式进行了归一化处理:
$$\overline{F_s} = \frac{F_s - \text{Min}(F_s)}{\text{Max}(F_s) - \text{Min}(F_s)} \quad (16)$$
式中 $\text{Max}(F_s)$ 和 $\text{Min}(F_s)$ 均是 E1、E2、E3 这3种情况的最大值和最小值。

从图2试验结果中可以看出,只考虑资源业务关联性 E2 和不考虑资源关联性 E1 的功能优化结果是一样的,也就是说资源的业务关联性并不影响企业的总成本。在算法刚开始运行时,三种情况的总成本相差不大,因为此时均是随机生成的资源配置。而随着算法的运行,资源的质量相关性的优势得到了极大的体现。在算法的收敛速度上,E2 和 E1 的优化配置效果在算法运行了50次左右基本达到了最优的结果,而 E3 在算法运行了25次左右就基本达到了 E2 和 E1 的最优配置效果。随着算法的运行,充分考虑资源的质量关联性,资源的优化配置效果还可以进一步优化,运行到40次左右,即达到了最优配置效果,并且最终的优化效果整体上均比不考虑关联性和只考虑业务关联性的情况好。可见,本文所提的考虑资源相关性的方法是有效可行的。

在试验过程中,相关参数的设置原则是尽快收敛到最优结果的同时尽可能的不出现局部最优情况。因此相关参数设置如下:根据候选资源集中资源最大数量设置种群规模 $\text{Size} = 100$,种群迭代最大次数为100次,交叉率为0.9,变异率为0.1,同时为了使得算法能够尽快收敛,依据压力值的取值范围,设置压力系数 $\chi = 0.9$ 。

试验是在以云企业一段时间内的总成本为目标,通过逐步算法迭代次数的增加,比较 E1、E2、E3 这三种情况下的总成本,试验结果如图2所示。

3.2 考虑资源相似性的试验及结果分析

在不考虑资源的关联性和先验性的情况下,只考虑资源质量相似性对最终目标函数的优化结果及算法效率来进行分析。对以下两种情况分别进行了试验:(1)不考虑资源相似性 E4;(2)考虑资源相似性 E5。

在试验过程中,相关参数设置如下:种群规模 $\text{Size} = 100$,种群迭代最大次数为100次,交叉率为0.9,变异率为0.1,压力系数 $\chi = 0.9$ 。

试验以云企业一段时间内的总成本(同样使用式(16)做归一化处理)为目标,通过逐步算法迭代次数的增加,比较 E4、E5 这两种情况下的总成本,试验结果如图3所示。

从图3试验结果中可以看出,考虑资源质量相似性 E5 的功能优化结果整体上要比不考虑资源质量相似性 E4 好。E4 的优化配置效果在算法运行了80次之后才达到了最优的结果,而 E5 在算法运行了50次左右就基本达到了最优配置效果。在算法刚开始运行时,两种情况的优化结果相差不大,因为此时均是随机生成的资源配置,而随着算法的运行,资源的质量相似性的优势慢慢体现出来。然而,到

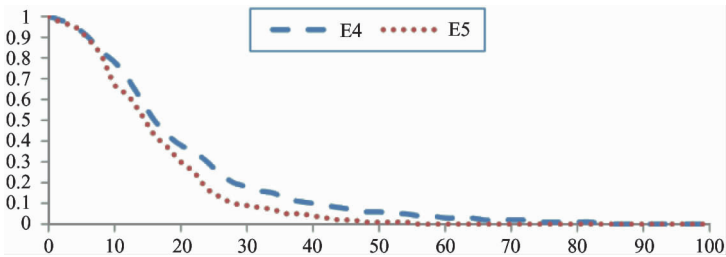


图 3 考虑资源相似性的试验结果

最后,考虑质量相似性与不考虑质量相似性的结果是趋于一致的,这是因为即使不考虑资源相似性,算法最终也能够通过足够的运算,最终找到相对较优的结果,只是收敛速度要慢一些。可见,本文所提的考虑资源相似性的方法是有效可行的。

3.3 考虑资源先验性的实验及结果分析

在不考虑资源的关联性和相似性的情况下,对只考虑资源先验性的最终目标函数的优化结果及算法效率来进行分析。对以下两种情况分别进行了试

验:(1) 不考虑资源先验性 E6;(2)考虑资源先验性 E7。

在试验过程中,相关参数设置如下:种群规模 $Size = 100$, 种群迭代最大次数为 100 次,交叉率为 0.9,变异率为 0.1,压力系数 $\chi = 0.9$ 。

试验以云企业一段时间内的总成本(同样使用式(16)做归一化处理)为目标,通过逐步算法迭代次数的增加,比较 E6、E7 这两种情况下的总成本,试验结果如图 4 所示。

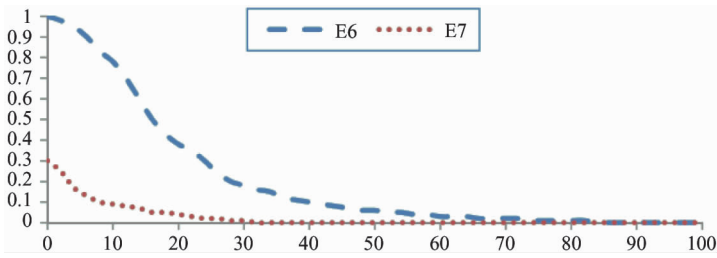


图 4 考虑资源先验性的试验结果

从图 4 实验结果中可以看出,考虑资源先验性 E7 的功能优化结果整体上要优于不考虑先验性 E6。E6 的优化配置效果在算法运行了 80 次之后才达到了最优的结果,而 E7 在算法运行了 30 次左右就基本达到了最优配置效果。在算法刚开始运行时,考虑资源先验性 E7 就比不考虑的功能优化结果好,因为这是先验性的性质决定的,经常被使用的资源往往是功能与性能均优的资源,在此优化的基础上,算法很快就收敛到最优的状态,但是到最后,不考虑先验性 E6 也能够通过足够的运算,最终达到与 E7 一样的优化结果,只是收敛速度要慢一些。可见,本文所提的考虑资源先验性的方法是有效可行的。

3.4 综合试验及结果分析

在以上分别考虑资源的 3 种特性的对比试验分析后,本节将综合考虑上述 3 种特性对最终目标函数的功能优化及算法效率进行分析。对以下 3 种情况分别进行了试验:(1)不考虑资源的 3 种特性的遗传算法 E8;(2)考虑资源的 3 种特性文献[13]的算法 E9;(3)考虑资源的 3 种特性本文的算法 E10。

在试验过程中,因为资源的 3 种特性使得算法相对复杂,可能导致收敛速度过慢,故参数设置与前述试验不同,详细参数设置如下:种群规模 $Size = 200$, 种群迭代最大次数为 200 次,交叉率为 0.95,变异率为 0.05,压力系数 $\chi = 0.95$ 。

试验以云企业一段时间内的总成本(同样使用式(16)做归一化处理)为目标,通过逐步算法迭代

次数的增加,比较 E8、E9 和 E10 这三种情况下的总

成本,试验结果如图 5 所示。

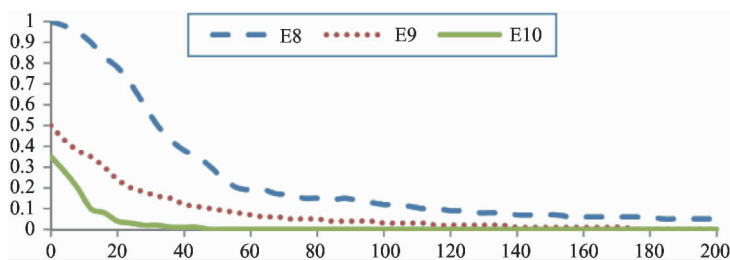


图 5 综合试验结果

从图 5 实验结果中可以看出,不考虑资源的 3 种特性的遗传算法 E8 的优化效果以及收敛速度都明显低于考虑 3 种特性的算法。而在考虑 3 种特性的两种算法中,本文所提的算法 E10 在算法初始阶段就要优于 E9,这是因为本文的算法在初始种群产生时就加入了资源的 3 种特性,而 E9 虽然也是考虑 3 种特性,但在初始化时,因粒子群个体的随机性,最多只能加入资源的先验性和相似性,而关联性则无法加入,只能在算法的运行过程中加入。而且在算法的收敛速度上,本文提出的算法要明显优于 E9,大约运行了 40 次左右就基本达到最优,而 E9 则运行了大概 180 次左右才能达到最优,但是二者的最优结果基本一致。可见,本文所提的算法在功能优化及算法效率上是有效可行的。

4 结 论

本研究在解决云企业的资源优化配置问题上,充分考虑了云制造环境下资源关联性、相似性及先验性等特点,分别建立了相应的计算模型,提出了资源的维持及使用策略,建立了云制造环境下云企业资源优化配置模型,然后采用基于实数矩阵编码的遗传算法,设计了适合实数矩阵编码的交叉变异规则。试验表明,本研究提出的方法都能够得到较好的资源优化配置结果和算法执行效率。本工作仅考虑了任务的交货期及成本两个因素,实际上还应扩展到更多的其他因素上,如资源的可靠性、信任度等;在算法上,解空间的编码上还存在着大量的无效解,可以通过优化编码进一步降低解空间的量级。以上都是下一步要研究的内容。

参考文献

- [1] 李伯虎,张霖,王时龙等. 云制造——面向服务的网络化制造模式. 计算机集成制造系统,2010,16(1): 1-7
- [2] 战德臣,赵曦滨,王顺强等. 面向制造及管理的集团企业云制造服务平台. 计算机集成制造系统,2011,17(3):487-494
- [3] Xu X. From cloud computing to cloud manufacturing. Robotics and computer-Integrated manufacturing, 2012, 28(1): 75-86
- [4] Lartigau J, Xu X F, Nie L S, et al. A restructuring service cluster algorithm ABC optimised based on virtual resource selection probability. International Journal of Computer Integrated Manufacturing, 2015, 28(9), doi: 10.1080/0951192x.2014.941405
- [5] Bansal S K, Bansal A, Blake M B. Trust-based dynamic web service composition using social network analysis. In: Proceedings of the 2010 IEEE International Workshop on Business Applications of Social Network Analysis (BASNA), Bangalore, India, 2010. 1-8
- [6] Mabrouk N B, Beauche S, Kuznetsova E, et al. QoS-aware service composition in dynamic service oriented environments. In: Proceedings of the 10th ACM/IFIP/USENIX International Conference on Middleware, Urbana, USA, 2009. 123-142
- [7] Lartigau J, Xu X F, Nie L S, et al. Similarity evaluation based on intuitionistic fuzzy set for service cluster selection as cloud service candidate. In: Proceedings of the 5th International IFIP Working Conference on Enterprise Interoperability (IWEI 2013), Enschede, Netherlands, 2013. 36-49
- [8] Amorim R, Claro D B, Lopes Det al. Improving Web service discovery by a functional and structural approach.

- In: Proceedings of the 2011 IEEE International Conference on Web Services (ICWS), Washington, DC, USA, 2011. 411-418
- [9] Tan W, Fan Y, Zhou M C, et al. Data-driven service composition in enterprise SOA solutions: a Petri net approach. *IEEE Transactions on Automation Science and Engineering*, 2010, 7(3): 686-694
- [10] Zhang Y, Lu J S, Ji X T. Research on strategies of networked manufacturing resources configuration based on evolutionary game. In: Proceedings of 2012 International Conference on Frontiers of Advanced Materials and Engineering Technology (FAMET), Xiamen, China, 2012. 1330-1334
- [11] Shen Y H, Yang X H. A self-optimizing QoS-aware service composition approach in a context sensitive environment. *Journal of Zhejiang University-Science C*, 2011, 12(3): 221-238
- [12] Alrifai M, Risse T. Combining global optimization with local selection for efficient QoS-aware service composition. In: Proceedings of the 18th international conference on World wide web, ACM New York, NY, USA, 2009. 881-890
- [13] Tao F, Zhao D, Hu Y, et al. Resource service composition and its optimal-selection based on particle swarm optimization in manufacturing grid system. *IEEE Transactions on Industrial Informatics*, 2008, 4(4): 315-327
- [14] El Hadad J, Manouvrier M, Rukoz M. TQoS: Transactional and QoS-aware selection algorithm for automatic Web service composition. *IEEE Transactions on Services Computing*, 2010, 3(1): 73-85
- [15] Cheng Z, Zhan D C, Zhao X B, et al. Multitask oriented virtual resource integration and optimal scheduling in cloud manufacturing. *Journal of Applied Mathematics*, 2014, 2014(Article ID 369350), 9 pages

Resource optimal configuration of cloud Enterprises in cloud Manufacturing

Cheng Zhen, Zhan Dechen

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001)

Abstract

The limitations of existing methods for resource optimal configuration of cloud enterprises, in cloud manufacturing are analyzed, and based on the consideration of the correlation, similarity and apriority among resources under cloud manufacturing environments, a new method for resource optimal configuration is presented from the perspective of the operating cost of cloud enterprises. The method defines the decision variables of the problem establishes the problem's model, then, models the correlation, similarity and apriority among resources and applies the corresponding models to the objective function, and finally, adopts the genetic algorithm based on the real number matrix to solve the problem of resource optimal configuration. The algorithm can approach the best solution more quickly because of the use of crossover and mutation operation rules suitable for the real number matrix as well as the genetic strategy with pressure calibration. The experimental results indicate that the proposed method efficiency, showing its feasibility and effectiveness.

Key words: cloud manufacturing, resource optimal configuration, correlation of resources, similarity of resources, apriority of resources