

基于过滤微粒群优化算法的虚拟企业伙伴选择

周伟¹, 卜艳萍²

(1. 华东理工大学 商学院, 上海 200237;

2. 上海交通大学 技术学院, 上海 201101)

摘要: 提出了一种过滤微粒群优化算法并应用于虚拟企业的伙伴选择问题。该算法以优良适应值微粒取代部分不良适应值微粒, 使算法具有过滤能力, 加快了搜索速度, 并保证收敛于全局最优解。仿真实验及与基本 PSO 算法的对比分析表明了 FPSO 算法的有效性。

关键词: 微粒群优化算法; 过滤; 虚拟企业; 伙伴选择

中图分类号: TP391

文献标识码: A

Partner selection of virtual enterprise based on filtering PSO algorithm

ZHOU Wei¹, BU Yan Ping²

(1. School of Business, East China University of Science and Technology, Shanghai 200237, China;

2. School of Technology, Shanghai Jiaotong University, Shanghai 201101, China)

Abstract: A filtering PSO (FPSO) algorithm is presented and applied to solve the partner selection problem of virtual enterprise. That is replacing the particles with worse fitness values by the feasible particles with better fitness values during the process of optimization. Thus, the algorithm owns the ability of filtrating, and the searching speed is improved, which can guarantee the global optimal solution being found as well. A series of experiments are performed and the simulation results are compared with standard PSO algorithm to show the efficiency of the proposed FPSO algorithm.

Key words: particle swarm optimization algorithm; filter; virtual enterprise; partner selection

在虚拟企业构建中, 合作伙伴的选择是一个关键的问题, 它直接关系到虚拟企业的成效。对合作伙伴的选择一般需要经过确定合作伙伴选择的目标、构建合作伙伴评价指标体系以及采用适当的优化方法等几个过程^[1]。分散网络化制造是新的生产模式, 按照这种新的制造原理而运作的经济实体称为制造网络联盟企业。由于制造网络联盟企业的动态性、协作性和分布性特点, 如何选择最佳的合作伙伴, 是组建制造网络联盟的关键。在联盟中, 最先发现市场机遇或需求, 并拥有主要核心资源的企业称为盟主, 盟主企业一般不具备独立完成全部生产的能力, 所以需要利用网络来进行伙伴的选择组成制造网络联盟^[2]。

微粒群优化 (PSO) 算法最早是由 Kennedy 和 Eberhart^[3]提出的, 它源于鸟群和鱼群群体觅食运动行为研究的启发, PSO 算法利用微粒之间的相互作用发现复杂问题解空间的最优候选区域。已经有许多学者用 PSO 算法解决组合优化问题^[4-5]。

本文将带有过滤机制的微粒群优化 (FPSO) 算法用《微型机与应用》2010 年第 02 期

于虚拟企业伙伴选择优化问题, 用 FPSO 寻找满足优化目标函数的最优企业组合方案。仿真实验表明 FPSO 算法用于求解动态联盟伙伴选择优化问题具有很好的效果, 得到了较基本 PSO 算法更为满意的结果。

1 问题描述

在盟主寻求合作伙伴时, 将任务分解成多个子任务, 这些子任务之间并不是独立的, 而是相互关联的^[6]。假设制造网络联盟需要由 n 项子任务组成:

$$Task = \{r_1, r_2, \dots, r_n\}$$

子任务 r_k 在制造网络中以节点来表示, 以下简称子任务 k 。每项子任务 k ($k=1, 2, \dots, n$), 都有 m_k 个候选伙伴参与竞标, 伙伴 i 对于子任务 k 的竞标价格为 c_{ki} , 承诺的加工时间为 t_{ki} 。

设可以完成子任务 k 的的候选企业集合为:

$$E_k = \{e_{kj} | j \in [1, m_k]\} (k=1, 2, \dots, n)$$

$$u_{kj} = \begin{cases} 1 & (\text{选择 } e_{kj} \text{ 参加制造网络联盟}) \\ 0 & (e_{kj} \text{ 未被选中}) \end{cases}$$

选择一组企业 $x=\{x_1, x_2, \dots, x_n\}$, 且满足 $x \cap E_i = \{x_i\}$ 。

如果子任务 k_1 必须在子任务 k_2 完工后才可以进行, 称这两个任务为连续任务对或关联任务, 记为 (k_1, k_2) , k_1 称为 k_2 的紧前子任务。

用 S_k 表示子任务 k 的开工时间, C_k 表示子任务 k 的完工时间。

当子任务 k_2 只有一个紧前子任务 k_1 时, 有:

$$S_{k_2} = C_{k_1}$$

当子任务 k_n 有多个紧前子任务 k_1, k_2, \dots, k_{n-1} 时, 则:

$$S_{k_n} = \max\{C_{k_1}, C_{k_2}, \dots, C_{k_{n-1}}\}$$

客户支付给盟主的总资金为 C_m , 根据项目进度情况分期支付给盟主。用 $e(t) \geq 0, (t=1, 2, \dots, D)$ 来表示客户分批支付给盟主的资金。若项目延期, 则盟主企业需要支付相对于总项目款的惩罚系数为 $\alpha (\alpha > 0)$ 的罚款。

为了使问题简化, 可以假设盟主将费用平均分两次支付给合作伙伴。第一次支付是在子任务 k 开始时, 另一次则在子任务 k 完成时支付。若合作伙伴未按期完成加工任务, 则需支付给盟主惩罚系数为 $\lambda (\lambda > 0)$ 的罚款。

盟主从客户那里得到的资金不一定能够满足子任务的需求, 可能引发资金短缺情况。资金短缺时, 盟主就要从银行贷款, 设银行的贷款利率为 $\beta (\beta > 0)$ 。

从效益的角度来分析, 联盟伙伴的选择的主要目的就是使整个订单的获益最高, 即最优的合作对象是使总成本最低。因此为了简化问题, 将完成项目的总成本定义为优化目标, 并定义若干个约束条件以保证项目按要求完成。

完成项目的总成本由以下几项组成:

$$Cost_1(x) = \sum_{i=1}^n c_{ik} \quad (1)$$

$$Cost_2(x) = \beta \times \sum_{i=1}^n \sum_{t=1}^{G_n(x)} \max\{0, [0.5 \times \sum_{S(x) \leq t} c_{ix_i} + 0.5 \times \sum_{C(x) \leq t} c_{ix_i} - \sum_{\tau=1}^t e(t)]\} \quad (2)$$

$$Cost_3(x) = -\lambda \times \sum_{i=1}^n \max\{0, [C_{iv}(x) - S_{iv}(x) - t_{iv}]\} \quad (3)$$

$$Cost_4(x) = \alpha \times \max\{0, [C_n(x) - D]\} \quad (4)$$

其中, x 表示一种制造联盟组合方案, D 为订单需求的完工日期, $Cost_i(x)$ 为第 i 项成本, x_k 为对应于子任务 k 选择的伙伴, $x_k \in [1, m_k]$ 。 $C_n(x)$ 为组成制造联盟 x 的完工日期, 式(1)表示按各个伙伴企业的竞标价格支付的总制造成本; 式(2)表示资金短缺情况下的贷款额; 式(3)表示合作伙伴未按期完成任务需要支付给盟主的罚款; 式(4)未按期完成项目支付给客户的罚款。

优化目标定义为:

$$\min Cost(x) = Cost_1(x) + Cost_2(x) + Cost_3(x) + Cost_4(x) \quad (5)$$

$$\begin{cases} S_{ix_i}(x) + t_{ix_i} = C_i(x) & i=1, 2, \dots, n \\ C_{ix_i}(x) \leq S_{kx_i}(x) & (i, k) \text{ 为关联子任务} \\ s.t. Cost(x) \leq C_m \\ \sum_{j=1}^{m_i} u_{kj} = 1 & k=1, 2, \dots, n \end{cases} \quad (6)$$

各项约束的含义依次为:

- (1) 每项子任务 i 完成时间 $C_i(x)$ 的约束;
- (2) 子任务之间的关联约束;
- (3) 所有子任务的加工成本之和 $Cost(x)$ 不高于项目的总投入 C_m ;

(4) 为每个子任务只选择一个候选企业作为制造合作伙伴。

由于该模型不是连续的可微函数, 而且可行解的范围也比较广, 所以用传统的数学方法难以得到有效的解, 因此采用过滤微粒群优化算法进行求解。

2 过滤微粒群优化算法在虚拟企业伙伴选择中的应用

2.1 微粒群优化算法

PSO 算法最初是模仿鸟群等群体动物寻找食物的社会性行为来建立的, 具有算法简单及容易实现的优点。在 PSO 算法中, 每个微粒都代表着搜索空间中的一个可行解, 所有微粒都有一个由被优化的函数决定的适应值, 微粒的速度和位置决定了它飞翔的距离和方向, 微粒通过跟踪两个极值来更新自己, 一个是微粒本身所经历的最优解, 称为个体极值 $pbest_i$; 另一个极值是整个种群目前找到的最优解, 称为全局极值 $gbest$ 。

假设在一个 m 维搜索空间中, 有 n 个微粒组成一微粒群, 其中第 i 个微粒的空间位置为 $X_i = (X_{i1}, X_{i2}, \dots, X_{im})$, 其飞行速度 $V_i = (V_{i1}, V_{i2}, \dots, V_{im}), i=1, 2, \dots, n$, X_i 是优化问题的一个潜在解, 将它代入优化目标函数可以计算出相应的适应值, 根据适应值可衡量 X_i 的优劣。

对每一代微粒, 其速度及位置的更新公式如下所示:

$$V_{i+1} = \omega V_i + c_1 R_1 (pbest_i - X_i) + c_2 R_2 (gbest - X_i) \quad (7)$$

$$X_{i+1} = X_i + V_{i+1} \quad (8)$$

式(7)中的 ω 为惯性权值, 它使微粒保持运动惯性。 c_1 和 c_2 都为正常数, 称为加速系数; R_1 和 R_2 是两个在 $[0, 1]$ 范围内变化的随机数, 用以保证微粒群体的多样性和搜索的随机性。

2.2 过滤微粒群优化算法

本文采用编码方式构造 n 维空间微粒的位置向量 X_i , 使它对应于有 n 个子任务的虚拟企业伙伴选择问题, 其中 X_i 的数值代表每个子任务的候选企业编号。例如由 8 个子任务组成的系统中, 每个子任务都有 4 个候选企业参与竞标。方案 $[2, 3, 1, 2, 3, 1, 4, 3]$ 表示第 1 个子任务由参与竞标的第 2 个候选企业完成, 第 2 个子任务由参与竞标的第 3 个候选企业完成, 第 3 个子任务由

《微型机与应用》2010 年第 02 期

应用奇葩

Example of Application

参与竞标的第 1 个候选企业完成,依次类推。

微粒按照式(7)、(8)进行更新迭代时,为了保证 X_i 的每一维数值为候选企业编号,每次迭代后需要对 X_i 进行向上取整处理。微粒的速度和位置都受最大值限制,速度的取值范围为 $[-(m_i-1), (m_i-1)]$,位置的取值范围为 $[1, m_i]$,其中 m_i 为子任务的候选企业数。在虚拟企业伙伴选择优化模型中,以(5)式表示的目标函数作为适应值函数。

过滤微粒群优化(FPSO)算法的基本思想是:首先计算出种群中每一个个体所对应的适应值,剔除适应值不良的部分个体,再从剩余的个体中选出一些适应值优良的个体,使这些个体在种群中复制一次,另外随机产生一些新个体,复制及新产生的个体数量之和等于过滤掉微粒的数量,以保持种群大小不变。这样,优良适应值的微粒取代部分适应值不良的个体,使得优秀适应值个体在种群中所占比例增大。为了避免搜索过程中过早地陷于局部最优解,适当补充新个体,增强种群活力,提高搜索到最优解的可能性。

用 FPSO 实现虚拟企业伙伴选择的算法流程为:

- (1) 随机产生具有 $popsiz$ 个微粒的初始种群,计算每个微粒的适应值,将第 i 个微粒的当前位置设置为该微粒的 $pbest_i$,将所有微粒中适应值最优的微粒位置设置为 $gbest$;
- (2) 根据公式(7)、公式(8)更新微粒,产生新一代种群。
- (3) 计算出每个微粒的适应值,剔除适应值不良的部分个体,选择适应值优良的部分微粒复制,同时产生一些新微粒,保持种群大小为 $popsiz$ 不变;
- (4) 如果微粒 i 的适应值优于自身的 $pbest_i$ 的适应值,则该微粒的 $pbest_i$ 更新为当前位置;如果某微粒的适应值优于 $gbest$ 的适应值,则 $gbest$ 更新为该微粒的当前位置;
- (5) 判断算法是否收敛到目标函数的最大迭代次数,如未满足条件,则转(2)继续搜索;
- (6) 输出全局最优适应值及其位置,将寻到的微粒最优位置,即最优任务选择方案赋给虚拟企业伙伴选择模型。

2.3 仿真实验及结果分析

在 MATLAB 环境下设计一个虚拟企业伙伴选择系统的模拟程序,该程序可以根据仿真的需要生成不同的任务数,各子任务的候选企业数,各个候选企业的性能指标等参数。在仿真中,独立进行 20 次实验,取最佳仿真结果。设初始种群为 60,进化代数数为 400。

以图 1 所示项目作为本次仿真的模型,图中给出了

各个子任务的时序关系。

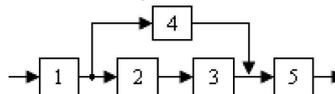


图 1 子任务间的时序关系

客户支付的总费用为 110 万元,工期为 24 周,资金支付情况如表 1 所示。如果项目超期,惩罚系数为 1.5 万元/周,盟主资金周转困难时的贷款利率为 0.02%/天。合作伙伴未按期完成加工任务的惩罚系数为 1.2 万元/周。5 个子任务分别有 4、3、5、5、4 个后选企业参与竞标。竞标情况如表 2 所示。

表 1 客户支付情况

| | | | | | | | | |
|--------|----|----|----|----|----|----|----|----|
| 时间/周 | 1 | 4 | 7 | 10 | 13 | 16 | 19 | 22 |
| 金额/3 元 | 20 | 15 | 12 | 12 | 12 | 12 | 12 | 15 |

表 2 候选企业的性能参数

| Sub-task | r_1 | | | | r_2 | | | | r_3 | | | | |
|------------|------------|----------|----------|----------|------------|----------|----------|----------|------------|----------|----------|----------|----------|
| | Enterprise | | | | Enterprise | | | | Enterprise | | | | |
| Enterprise | e_{11} | e_{12} | e_{13} | e_{14} | e_{21} | e_{22} | e_{23} | e_{24} | e_{31} | e_{32} | e_{33} | e_{34} | e_{35} |
| Cost | 19.5 | 20.8 | 18.2 | 17.5 | 21.8 | 19.5 | 20.1 | 18.3 | 17.5 | 10.8 | 19.5 | 20.5 | |
| Time | 6 | 5 | 7 | 8 | 5 | 7 | 6 | 6 | 8 | 7 | 5 | 4 | |
| Sub-task | r_4 | | | | r_5 | | | | | | | | |
| | Enterprise | | | | Enterprise | | | | | | | | |
| Enterprise | e_{41} | e_{42} | e_{43} | e_{44} | e_{51} | e_{52} | e_{53} | e_{54} | | | | | |
| Cost | 17.0 | 18.2 | 21.5 | 20.8 | 16.0 | 20.5 | 19.5 | 18.0 | | | | | |
| Time | 8 | 7 | 5 | 6 | 9 | 5 | 6 | 7 | | | | | |

假设盟主在每个子任务开始时支付总金额的 50% 作为启动资金,剩余的 50% 在子任务完成时支付。先用穷举法对该项目进行最优伙伴选择,选择结果及寻优时间作为以下算法应用的参照。将 FPSO 和 PSO 算法同时用于该模型的伙伴选择,图 2 和图 3 分别表示采用 FPSO 和 PSO 算法时目标函数的收敛情况。

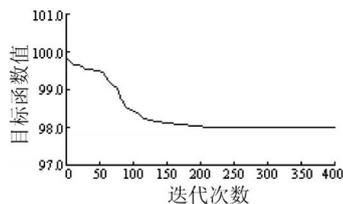


图 2 FPSO 的搜索结果

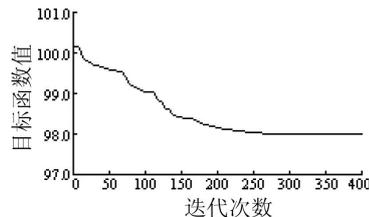


图 3 PSO 的搜索结果

从仿真结果可以看出,采用 FPSO 进行伙伴选择寻优时的平均迭代次数为 202,采用 PSO 算法时的平均迭

(下转第 98 页)

代次数是 264, FPSO 算法比 PSO 算法节省了 23.48% 的寻优时间, 因此, 在虚拟企业伙伴选择模型应用中, FPSO 算法优于 PSO 算法。另外, 与穷举法相比采用 FPSO 算法和 PSO 算法都能较快地找到最优候选企业组合。当改变子任务数量及子任务间的时序关系时, FPSO 算法的优越性更明显。

本文提出了一种带过滤机制的 PSO 算法, 并应用于制造网络联盟企业的伙伴选择模型优化。FPSO 算法在进化过程中, 通过过滤掉适应值较差的部分微粒, 而保持种群的优良特性和活力, 从而保证了搜索到最优解的可能性。与基本 PSO 算法进行对比仿真研究, 结果表明 FPSO 算法能较快地搜索到最优解。

参考文献:

- [1] IP W H, HUANG M, YUNG K L, et al. Genetic algorithm solution for a risk-based partner selection problem in a virtual enterprise [J]. Computers & Operations Research, 2003, 30: 213-231.
- [2] CAO H Y, WANG DW. A simulation based genetic algorithm for risk-based partner selection in new product development [J]. International Journal of Industrial Engineering, 2003, 10 (1): 16-25.
- [3] KENNEDY J, EBERHART R C. Particle swarm optimization [C]. Proceedings of IEEE International Conference on Neural Networks. Australia: IEEE Service Center, 1995: 1942-1948.
- [4] CLERC M. Discrete particle swarm optimization [J]. In: Onwubolu

G C, Babu B V, eds. New Optimization Techniques in Engineering. Berlin, Germany: Springer-Verlag, 2004: 219-240.

- [5] CAGNONA L, ESQUIVEL S, GALLARD R. Particle swarm optimization for sequencing problems: A case study [J]. In: Proc of the Congress on Evolutionary Computation. Oregon, Portland, 2004(1): 536-541.
- [6] WU N Q, SU P. Selection of partners in virtual enterprise paradigm [J]. Robotics and Computer-Integrated Manufacturing, 2005, (21): 119-131.

(收稿日期: 2009-09-24)

作者简介:

周伟, 男, 1966 年生, 副教授, 主要研究方向: 供应链优化及电子商务。

卜艳萍, 女, 1964 年生, 副教授, 主要研究方向: 供应链优化及算法研究。