

新产品开发合作伙伴的优化选择研究

曹 洪 医

(中南财经政法大学 工商管理学院, 湖北 武汉 430060)

摘 要: 研究新产品开发项目中的合作伙伴选择问题, 用带有非线性目标函数的 0-1 整数规划模型对问题进行定量描述, 在时间和成本约束下的最小化项目失败风险。模型的非线性和复杂性使其无法用常规方法进行求解, 设计带有自适应惩罚函数的遗传算法对可行域和非可行域进行搜索, 快速有效地获得最优解或近优解。

关键词: 伙伴选择; 新产品开发; 遗传算法; 自适应惩罚函数

中图分类号: F403.6

文献标识码: A

文章编号: 1001-7348(2006)12-0061-03

0 前言

在今天的全球化制造环境下, 生产技术迅速发展, 系统需求不断扩展, 使得在新产品开发中通过策略联盟获得技术优势变为重要手段。进行策略联盟的目的在于实现风险共担、获得成本优势、发挥市场潜力, 或者, 获得令客户满意的技术等^[1]。

敏捷制造环境下的全球性伙伴关系可以用 Internet 驱动的招投标进行描述^[2]。假设一个全球性企业启动某个新产品开发项目, 将该项目划分为若干开发任务后, 通过招投标为其中部分任务选择开发合作伙伴。虽然有很多因素影响伙伴选择, 例如信誉、可靠性、公司关系等, 但本文将关注点集中在成本、时间和风险 3 个要素上。由于合作伙伴的候选企业在上述 3 个因素上可能存在很大差异, 需要综合考虑以便进行优化。

在敏捷制造和供应链管理环境下, 伙伴选择问题受到广泛关注^[3,4], 因为它是扩展—虚拟企业信息管理系统的一个主要功能^[5,6]。但是, 如何为伙伴选择构建数学模型仍是运筹学领域的一项挑战^[4,7]。1996 年, Tulluri 等提出了伙伴选择的两阶段数学模型^[8], 用活动网络关系形式描述多个任务之间的时序关系, 使得问题成为内含项目调度的伙伴选择问题。

本文用非线性 0-1 整数规划模型来描述新产品开发中的伙伴选择问题, 模型的目标是在成本和时间的约束下, 最小化项目总体开发失败风险。由于模型的复杂性和非线性, 难以用常规方法求解, 因此设计有针对性对搜索历史自动调整的惩罚函数的遗传算法, 对可行域和非可行域进行搜索。通过多个不同规模数值例子的仿真证明了算法的效率。

1 问题描述与模型建立

假定新产品开发项目由 N 个开发任务组成, 由于该全球性企业不具备项目开发所需要的全部能力, 决定为其中的某些任务选择开发合作伙伴。在新产品开发中, 合作伙伴的候选企业很多, 包括其它大型公司、研究机构、政府实验室、高科技公司等^[9]。项目开发任务以活动网络形式进行组织, 本文考虑有关活动网络的一般性假设^[9]。

讨论新产品开发成功问题的文献很多, 诸多经验研究表明了新产品开发具有较高的失败率, 很多新产品开发项目的失败在于没有对开发风险进行有效的辨识和管理^[10]。

定义两类事件。事件 S 表示项目开发成功。事件 A_i 表示第 i 个开发活动成功, i 为活动编号, $i=1, \dots, N$ 。基于事件运算和概率理论, 当项目中的所有开发活动相互独立时,

项目成功概率 $P\{S\} = \prod_{i=1}^N P\{A_i\}$, 此公式的证明可参见文献[11]。其中, $P\{\cdot\}$ 表示事件的概率。

若活动 i 是活动 k 的紧前活动, 则称 (k, i) 为连续活动对, $i, k=1, \dots, N$ 。所有连接活动对的集合为 H。对于活动 i, 合作伙伴的候选企业为 m_i 个, $\forall i$ 。对于 i 的第 j 个候选企业, 其竞标成本和开发时间分别 b_{ij} 和 d_{ij} , $\forall i, j=1, \dots, m_i$ 。为了保证新开发的产品按时投入市场, 定义 D, 表示项目的最终期限。为了控制新产品开发的成本, 定义 B, 表示该全球性企业对项目的最高投入。

可以通过 Delphi 等方法^[12], 在综合考虑多方因素的基础上, 利用专家经验, 估计每个候选企业的失败概率。对于活动 i 的第 j 个候选企业, 其失败概率为 r_{ij} , $\forall i, j$ 。

定义决策变量:

$$w_{ij}(t) = \begin{cases} 1 & \text{活动 } i \text{ 选择候选企业 } j \text{ 且在 } t \text{ 开始} \\ 0 & \text{其它} \end{cases}, \forall i, j \quad (1)$$

新产品开发伙伴选择问题可以用下列模型进行描述。

$$\max_w Z(w) = \prod_{i=1}^N \sum_{j=1}^{m_i} \sum_{t=1}^d (1 - w_{ij}(t)r_{ij}) \quad (2)$$

$$\text{s.t. } \sum_{i=1}^N \sum_{j=1}^m \sum_{t=1}^d w_{ij}(t)b_{ij} \leq B \quad (3)$$

$$\sum_{t=1}^d \sum_{j=1}^m w_{ij}(t)(t+d_{ij}) \leq \sum_{t=1}^d \sum_{j=1}^m w_{kj}(t)t, \quad \forall (i, k) \in H \quad (4)$$

$$\max_i \left\{ \sum_{j=1}^m \sum_{t=1}^d (t+d_{ij})w_{ij}(t) \right\} = d \quad (5)$$

$$d \leq D \quad (6)$$

$$w_{ij}(t) = 0 \text{ 或 } \forall i, k, t \quad (7)$$

$$\sum_{j=1}^m \sum_{t=1}^d w_{ij}(t) = 1, \quad \forall i \quad (8)$$

通过目标 $Z(w)$ 最大化项目开发成功率。约束(3)表示所有活动的开发成本之和并不高于项目最高投入。约束(4)表明开发项目不能在最终期限之后完成。约束(5)和(6)用来描述项目中活动的顺序关系。约束(8)保证为每个活动只选择一个候选企业作为开发合作伙伴。

由于模型的非线性和复杂性,难以用常规方法进行求解,在下一部分中提出带有自适应惩罚函数的遗传算法。

2 遗传算法

在遗传算法的构建过程中,必须为其设计适当的编码方式、适应函数、选择策略、遗传算子和停止准则^[13,14]。在对多个数值问题进行仿真时,发现模型(2-8)所描述的这类伙伴选择问题具有很强的约束性,因此本文在遗传算法的应用中着重研究如何对非可行解进行惩罚,并提出自适应惩罚函数。

2.1 编码和模型简化

使用一串自然数作为遗传算法中的基因表达。令向量 $x=[x_1, \dots, x_N]$, x_i 是 1 和 m_i 之间的整数,表示为活动 i 选择第 x_i 个候选企业作为其合作伙伴, $\forall i$ 。这样, $x=[x_1, \dots, x_N]$ 称为一个选择,代表了遗传运算中的一条染色体。

对于一个选择 x , $e(x)$ 和 $c(x)$ 为选择活动 i 的开始时间和结束时间, $\forall i$, 二者可能通过项目计划计算出来^[15]。假定新产品开发项目的计划策略使所有活动尽早开始,则通过公式(9)可计算出活动 i 的最早开始时间,最早结束时间 $c_k(x)=e(x)+d_{k,x}, \forall i$ 。

$$e(x) = \begin{cases} \max\{c_k(x)\} & \exists (k, i) \in H \\ 1 & \text{其它} \end{cases} \quad (9)$$

模型(2-8)可以简化为如下模型。

$$\max_w Z(w) = \prod_{i=1}^N \sum_{j=1}^m \sum_{t=1}^d (1-r_{ix}) \quad (10)$$

$$\text{s.t. } \sum_{i=1}^N b_{ix} \leq B \quad (11)$$

$$c(x) \leq e(x), \quad \forall (i, k) \in H \quad (12)$$

$$\max_i \{c_i(x)\} = d \quad (13)$$

$$d \leq D \quad (14)$$

$$x_i = 1, \dots, m_i, \quad \forall i \quad (15)$$

2.2 惩罚函数

对带约束的问题而言,当遗传算法对多个染色体进行交叉运算和变异运算时,很难保持其可行性。即使对两个可行的双亲进行交叉运算,也可能产生非可行的后代。对于采用非二进制编码的组合优化问题而言,更是如此^[16]。

为了产生可行后代,可以采用“修复算子”。但是,对于很多优化问题而言(也包括本文所研究的伙伴选择问题),这种方法并不适用。修复运算可能十分复杂,或者,需要更多的计算时间。而在某些情况下,寻找可行解本身就是 NP-难问题。此时,可采用外部惩罚方法^[16]。

惩罚函数在遗传算法中的应用非常普遍,并形成几种重要方法。最基本的惩罚方法是严厉惩罚,产生非可行解时立即将其消除。也可以基于所违反的约束的数量构造惩罚函数,而不考虑违反约束的程度。另外,还可基于非可行解与可行域之间的距离构造惩罚函数^[16]。

采用基于非可行解与可行域的距离构造惩罚函数时,如果随着搜索过程改变对给定距离的惩罚(通常随着迭代增加),则使得这种惩罚函数具有了动态特性。这既有利于在遗传搜索的初始阶段允许在更广泛的非可行域进行搜索,又保证了搜索最终收敛到可行域中^[16]。虽然此类惩罚函数是动态的,但其并不针对搜索历史中的信息进行调整,而仅仅根据代数进行单调的增加。有些学者提出利用搜索历史信息进行自动调整的惩罚函数,有利于增强惩罚函数的稳健性和有效性,通用形式可参见文献[17]。

本文基于 Coit 和 Smith 的工作构造自适应惩罚函数^[16]。用 NFT 表示每个约束的“近可行性阈值”。对于给定染色体,外部惩罚函数通常被描述为到可行域的距离的非减函数。NFT 是一个与可行域距离有关的阈值,可基于约束违反的实际情况对其进行选择。惩罚函数鼓励在可行域和距离小于 BFT 的非可行域内进行搜索,同时也允许在其它

非可行域内进行搜索。

惩罚函数试图基于对特定问题约束的违反程度自动进行调整。惩罚函数由两项构成:违反约束的相对惩罚系数,自适应惩罚项。其表达式见公式(16)。

$$V_{lp} = V_{l-} \left[\left(\frac{\Delta b_i}{NFT_b} \right)^\beta + \left(\frac{\Delta d_i}{NFT_d} \right)^\beta \right] (V_{ai} - V_{feas}) \quad (16)$$

$(V_{ai} - V_{feas})$ 称为自适应项,其中, V_{ai} 为到目前为止搜索到的未受惩罚最好解的目标函数值, V_{feas} 为到目前为止搜索到的最好可行解的目标函数值。

对于某个解 l , V_l 和 V_{lp} 分别代表目标函数值和施加了惩罚后的目标函数值。幂次 β 是预先设定的参数,用来调整惩罚的严厉性。对于伙伴选择问题, NFT_b 和 NFT_d 分别表示项目成本和时间约束的“近可行性阈值”。 Δb_i 和 Δd_i 分别表示解 l 违反上述时间和成本约束的程度。

需要注意的是,自适应惩罚项 $(V_{ai} - V_{feas})$ 可能带来两个问题:零惩罚和过惩罚。当 $V_{ai} = V_{feas}$ 时,即使有非可行解,对所有非可行解也给予零惩罚。而在遗传搜索初始阶段,当某个不可行解的目标函数值很大时, V_{ai} 则很大,会给所有非可行解给予过度惩罚。

针对伙伴选择问题中给定的成本和时间约束,在多大的非可行域中进行搜索并不非常直观。可通过对问题的特定算例的迭代实验确定 NFT 的合适值。为了解决这个问题,可使用随着迭代过程递减的动态 NFT,见公式(17)。

$$NFT = \frac{NFT_0}{1 + \lambda g} \quad (17)$$

NFT_0 是 NFT 的上限(或称初始值)。 g 是代数。 λ 是一个常数,用来保证对 NFT_0 和 0 之间的整个区域进行搜索。通过设定 λ 值,避免动态 NFT 太快或太慢达到 0。

2.3 遗传运算

常用的交叉算子为单断点交叉、双断点交叉和均匀交叉,本文采用均匀交叉,因其对于组合优化问题具有较好的性能^[19]。基于双亲在种群中排序结果的平方值确定选择概率,这种方法的优点是计算效率很高,不仅保证遗传搜索的选择压力,也避免遗传搜索陷入局优^[20]。采用随机摄动变异作为变异算子。

在交叉运算之后,将当前代和新产生的后代进行组织,产生下一代,之后进行变异

采用精英策略,将到目前为止获得的最好解保存起来,并始终不被选择进行变异以确保算法的收敛性^[20]。

2.4 停止准则

对于遗传算法,一般采用3种停止准则,最大代数、执行时间和适值收敛性^[20]。本文选择最大代数作为停止准则,当算法达到某一预先的最大代数时,停止搜索。

3 计算结果

算法用 Visual C++ 语言编制在 PC/586 上运行。为了测试遗传算法的性能,随机产生伙伴选择问题的若干个不同规模的计算实例。分别对这些问题应用3类惩罚函数,严厉惩罚、常数 NFT 惩罚和动态 NFT 惩罚。通过对3种方法的比较,发现常数 NFT 惩罚函数仅具有适应性,而动态 NFT 惩罚函数则兼具适应性和动态性(随着代数增加惩罚力度)。

虽然动态方法并没有在所有计算实例中都表现出很强的性能,但是统计结果表明其具有普遍的优越性,这种优越性随着约束运气程度的增加而增加。附表中给出汇总表,对于特定问题应用动态 NFT 惩罚函数时,所获得的最终解对于惩罚参数、种群数、约束程度、特定计算实例等具有很强的稳健性。

附表 计算实例的可行性和性能比较

比较	可行解的百分比	达优率
严厉惩罚	100%	20%
3% NFT	80%	83%
动态 NFT	100%	90%

4 结束语

本文的研究工作可总结为下述两点。首先,模型(2-8)对新产品开发项目的合作伙伴选择问题进行了较为正式的描述,在动态竞争的全球环境中,可以进一步扩展其复杂

性。其次,带有自适应惩罚函数的遗传算法适用于对此类问题进行求解,对于大规模和较强约束的计算实例体现出较强的性能。

参考文献:

[1] R.N.Nagel, and R.Dove, 21st century manufacturing enterprise strategy: and industry-lead view, Bethelam, PA: The Iacocca Institute, Lehigh University,1991.

[2] A. Gunasekaran. Agile manufacturing: enables and an implementation framework. International Journal of Production Research,1998,36,(5):1223 - 1247.

[3] S.Talluri, and R.C.Baker. A quantitative framework for designing efficient business process alliances. Proceeding of IEEE International Engineering Management Conference, Managing virtual enterprise:a convergence of communications, computing, and energy technologies, Piscataway, NJ,1996,656- 661.

[4] H.Y. Cao, Research on models and optimization of risk control in dynamic alliance of enterprise, Ph.D. Thesis, Shenyang, PRC: Northeastern University,2002.

[5] S.L. Goldman, R.N. Nagel, and K.Preiss, Agile competitors and virtual organizations, New York: Van Nostrand Reinhold,1995.

[6] Z.Su, and D.Poulin. Partnership management within the virtual enterprise in a network. Proceeding of IEEE International Engineering Management Conference, Piscataway,NJ,1996,645- 650.

[7] M.J.Maloni, and W.C.Benton. Supply chain partnerships: opportunities for operations research. European Journal of Operation research,1997,101, (3):419- 429.

[8] M.Rice, G.C.O'Connor, L.Peters, and J.Mmorone. Managing discontinuous innovation.

[9] S.E.Elmaghraby, Activity networks-project planning and control by network models, New York: John Wiley and Sons,1997.

[10] R.Ahmedi, and Wan, H. Managing Development

Risk in Product Design Processes. Operations Research,1999,47,235- 246.

[11] H.Y.Cao, and D.W.Wang. A simulation based genetic algorithm for risk-based partner selection in new product development. International Journal of Industrial Engineering,2003,10,(1):16 - 25.

[12] E.P.Sioshansi. Subjective evaluation using expert judgement: an application. IEEE Transactions on Systems,Man and Cybernetics,1983,(3): 391- 397.

[13] J.Holland, Adaptation in Natural and Artificial Systems, Ann Arbor: University of Michigan Press,1975.

[14] Z.Michalewicz, Genetic algorithms + data structure=evolutionary programming, New York:Springer,1994.

[15] P. Brucker, A. Drexl, R.Mohring, K.Neumann, and E.Pesch.Resource-constrained project scheduling:notation,classification, models, and methods. European Journal of Operations Research, 1999,112,(1):3- 41.

[16] Z.Michalewicz. A survey of constraint handling techniques in evolutionary computation methods. Proceedings of the Fourth Annual Conference on Evolutionary Programming, Cambridge,MA:MIT Press,1995,135- 155.

[17] D.W.Coit, and A.E.Smith. Unequal area feasibility layout using genetic search. IIE Transactions,1995,27:465- 472.

[18] D.W.Coit and A.E.Smith. Penalty guided genetic search for reliability design optimization. Computers & Industrial Engineering, 1996, 20,(4): 895- 904.

[19] G.Syswerda. Uniform crossover in genetic algorithms. Proceedings of the Third International Conference on Genetic Algorithms, San Mateo, CA: Morgan Kaufmann Publishers,1989,2- 9.

[20] M.Gen and R.Cheng. Genetic algorithm and engineering desing. New York: Wiley & Sons,1997.

(责任编辑:董小玉)

Research on Optimal Partner Selection in New Product Development

Abstract: In this paper, we investigate the problem of partner selection in new product development, First, we give a formal description of the problem and an integer programming model with nonlinear objective function to minimize project failure risk within the constraints of cost and time.

Key words: pannerselection; new product development; genetic algorithm; adaptive penalty function