

基于自适应算子的混合进化算法及其应用

游晓明¹ 刘升¹ 帅典勋²

(上海工程技术大学电子电气工程学院 上海 200065)¹

(华东理工大学计算机科学与技术系 上海 200237)²

摘要 提出了一种求解多目标优化最短路径问题的混合进化算法。算法中依据小生境机制生成若干个实数编码染色体的子群,各子群分别利用自适应算子的局域搜索能力找出优化解。协同进化机制能更好地保证进化的方向性和种群的多样性,基于路径表示的染色体十进制编码方法以及染色体的交叉和变异具有新颖性。该算法用于解决智能交通系统的公共交通线路换乘问题,实验结果表明了其优越性。还运用 Markov 随机过程理论证明了算法的收敛性。

关键词 进化算法,自适应算子,十进制编码染色体,最短路径,小生境

中图分类号 TP301.6 **文献标识码** A

Hybrid Evolutionary Algorithm Based on Adaptive Operator and its Application

YOU Xiao-ming¹ LIU Sheng¹ SHUAI Dian-xun²

(College of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 200065, China)¹

(Dept. of Computer Science and Technology, East China University of Science and Technology, Shanghai 200237, China)²

Abstract A novel hybrid evolutionary algorithm based on adaptive operator for solving multi-objective optimization was proposed. By niche methods population is divided into subpopulations of real-coded chromosome automatically, each subpopulation can obtain optimal solution by self-adaptive mechanism. We introduced real-coded chromosome with innovation to solve precision and efficiency problem of binary system; co-evolutionary strategy of niche can guarantee quite nicely the population diversity and the convergence speed. The algorithm is applied to urban public transportation system transfer, and experimental results show its superiority. The convergence of the algorithm is proved based on Markov chain in this paper.

Keywords Evolutionary algorithm, Self-adaptive operator, Real-coded chromosome, Shortest path, Niche

智能交通系统(ITS)是目前世界交通运输领域研究的前沿问题^[1,10]。它能有效提高车辆运行效率,改善行车安全,节约能源,减少交通阻塞,减轻环境污染,是目前国际公认的解决城市和公路交通问题的最佳途径。其中的许多问题属于强 NP 难题,因此探索近似算法是必要和现实的。本文尝试采用进化算法对 ITS 问题中的交通线路换乘多目标优化问题进行求解。进化计算求解问题的基本思想相当简单^[2,4]:由问题的候选解组成一个种群,然后通过随机变化和选择等算子进行演化。其中随机变化决定了发现新解的机制,选择则确定保持哪些解作为下一步搜索的基础。从演化计算产生至今,算法设计一直是它的核心,也是人们所关注和研究的重点。随着 ITS 的逐步开发和推广,对大规模路网中交通分配的实时性提出了很高的要求,而最短路径的求解是公共交通线路换乘中最基本也是最重要的环节,如何保持算法的高效和稳定性,一直是关注的热点和难点。

本文以求解 ITS 智能公交信息服务子系统中的公共交通线路换乘多目标优化问题为例,分析了进化算法的全局优

化能力和局域搜索能力,综合各种进化操作算子的特点,提出了一种求解基于多目标优化最短路径问题的自适应混合进化算法。算法中依据小生境机制生成 m 个实数编码染色体的子群组 $G_i(0)(i=1, 2, \dots, m)$, $G_i(0)$ 中的染色体编码表示如下:“ $SX_2X_3 \dots X_{j-1}D$ ”,其中第一个位置 S 总是路径的源节点,最后一个位置 D 是目标节点,染色体编码中不允许有重复的基因位。每个子群组 $G_i(0)$ 采用协同进化机制并利用自适应算子的局域搜索能力可找出优化解。我们不仅用 Markov 随机过程理论证明了该进化算法的收敛性,而且通过公共交通线路换乘问题的实验结果说明了该进化算法的优越性。

1 混合进化算法及其收敛性

1.1 算法描述

传统进化算法中的二进制染色体表达不适合多参数优化和高精度计算,它与十进制转换繁琐,并且进化的方向性无法有效控制。为此,引入了一个自适应操作算子^[3,5],提出了一种基于路径表示的染色体十进制编码方法,并依据小生境机

到稿日期:2009-03-30 返修日期:2009-06-08 本文受国家自然科学基金项目(资助号:60575040),上海市自然科学基金项目(资助号:09ZR1420800),大学生创新项目(资助号:cx0902006)资助。

游晓明(1963—),女,博士,副教授,主要研究方向为人工智能、分布并行计算,E-mail:yxm6301@163.com;帅典勋(1941—),男,博士,教授,博士生导师,主要研究方向为人工智能、分布并行计算、人工生命、网络行为动力学。

制给出求解大规模路网中公共交通线路换乘优化问题的自适应混合进化算法(TRHEA),其结构可描述如下:

TRHEA()

```
{t=0;
  初始 Q(0)为路径备选集;
  Q(0) = {"SX2X3...Xj-1Xj"或"X1X2X3...Xj-1D"};
  依据小生境方法将 Q(0)划分为 m 个子群组 Gi(0)
  {生成 k 个实数编码染色体:"SX2X3...Xj-1D"}
  For 每个子群组 Gi(0)
    While(not termination-condition)do
      {t=t+1;
        对 Gi(t)实施自适应操作算子:
        { 染色体交叉;
          染色体变异;
          染色体的自适应促进与抑制;
          记忆个体产生;}
        评价 Gi(t);
        保存 Gi(t)中的最优解;
        子群组 Gi(t)间协同进化;
      }
    }
```

1.2 染色体编码

在进化算法中,首先要确定问题的编码。在该算法中,染色体由一系列的整数序列组成,即基于路径表示的编码方法,此方法是一种最自然、最简单的表示方法。要求每个个体的染色体编码中不允许有重复的基因位,也就是说要满足任意一个站点在路径中只能访问一次的约束。染色体的第一个位置总是路径的源节点,最后一个位置是目标节点,染色体的长度是变化的,染色体编码即由源节点到目的节点的序列组成。

初始化 Q(0)为路径备选集时,所要配置的每一条路径 L_i ($1 \leq i \leq \max$),是从源节点 S 到目的节点 D 的可选路径集, \max 为所要配置的最大路径数目。 $\{L_i = X_1 X_2 \dots X_j\}$ 作为进化算法编码空间的备选路径集,其中 X_1 为 S 或 X_j 为 D,整个染色体编码为 "SX₂X₃...X_{j-1}D",如图 1 所示。染色体编码的第一位置基因是源节点,第二位置基因是从与源节点连接的其它节点中随机选择或启发选择的。重复这个过程,直至找到目的节点。

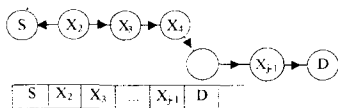


图 1 染色体编码

1.3 自适应操作算子

自适应操作算子包括染色体交叉、变异、染色体的促进与抑制、记忆个体生成,其中,记忆个体生成算子的作用在于保证自适应系统中进化进程的方向性,它充分利用局部最优解来生成适应度更好的染色体。交叉和变异算子属于全局搜索算子,其作用在于保持多样性的染色体群体,扩大搜索范围,原则上可扩大到整个搜索空间。

染色体交叉:交叉采用相同路径保留的方法,假设给定两个父代个体 T_f 和 T_m ,通过交叉操作将生成子代个体 T_c ,则其相同的路径部分应该是造成它们适应度高的因素,代表了父代身上具有的优秀特征,应该遗传给后代。而对于 T_f 和 T_m 中不相同的路径部分,则从备选集中重新选择其它解。如图 2(a)所示,两个父代个体中潜在的交叉位置为 9 和 15,随机选择位置 9 作为交叉位置,通过交换子路径得到新的子代

个体。若 T_f 和 T_m 中没有相同的站节点,则选取在同一地铁站可以换乘的两个公交站节点作为交叉位置。



图 2

染色体变异:变异操作是一个父代个体产生一个子代个体。从当前个体的染色体基因中随机选择变异节点位置 j ,以完成实数表达的染色体的变异操作。具体实现如下:

设当前个体的染色体为 $X_1 X_2 \dots X_j \dots X_k$,利用混沌随机变量选择变异节点位置 j ,然后在备选集染色体中选择 $X_j X'_{j+1} \dots X'_{l-1} X_k$,得到新的个体 $X_1 X_2 \dots X_j X'_{j+1} \dots X'_{l-1} X_k$ 。如图 2(b)所示,生成的随机数为 5,当前个体的第 5 节点位置 16 作为变异点,当前个体的染色体中选择节点 1 到目的节点 16 的子路径 1(1 3 8 11 16),在备选集中选择节点 16 到目的节点 21 的子路径 2(16 19 21),连接这两个子路径,完成变异操作。

算法中的染色体变异率大小可由染色体的亲和度高低来调节,具体分为两类:高亲和度染色体和低亲和度染色体,变异率依次为 P_l, P_h ,且 $P_l < P_h$ 。于是,高亲和度的染色体作为记忆个体能够继续搜索以进一步提高亲和度,而低亲和度的染色体能够保持种群多样性,扩大搜索范围。这种变异方式体现了负反馈机理,即变异后,若提高或降低了染色体的亲和度,则在下一轮变异时,亲和度将反作用于变异率,从而使进化过程具有稳定性。

染色体自适应促进抑制是指利用克隆选择机制产生记忆个体。当有亲和度更高的染色体产生时,则必须替换低亲和度的染色体,而为保持种群的多样性,具有相对较低亲和度的染色体以较大的变异率进行变异,从而被抑制。

1.4 协同进化算子

子群组 $G_i(t)$ 间交换最优模式来达到协同进化的目的。 $G_i(t)$ 中的个体选自路径备选集, $G_i(t)$ 进化后的优秀个体替代 $G_{i+1}(t)$ 进化前的部分个体, $i=2,3,\dots,m-1$ 。协同进化算子既能更好地保证进化的方向性又能维持种群的多样性,使得进化过程更加稳定。

1.5 算法收敛性

定理 1 求解交通线路换乘优化问题的实数编码自适应混合进化算法(TRHEA)的种群序列 $\{A(n), n \geq 0\}$ 是有限齐次马尔可夫链。

证明:TRHEA 算法的种群状态转移情况可以表示成如下的随机过程: $A(K) \xrightarrow{\text{cross}} A1(K) \xrightarrow{\text{mutation}} A2(K) \xrightarrow{\text{select}} A3(K) \xrightarrow{\text{exchange pattern}} A(K+1)$ 。

TRHEA 采用实数编码染色体 $X_1 X_2 \dots X_j \dots X_k$,由于 X_i 的取值是连续的,因此理论上种群所在的状态空间是无限的;但另一方面,实际运算中 X_i 是有限精度的,设 X_i 的精度是 ϵ (例如: ϵ 可为 10^{-5} 或 10^{-6}),则其维数为 $V = (H-L)/\epsilon$, H 是 X_i 取值的上限, L 是 X_i 取值的下限,假设染色体长度为 M ,种群规模为 N ,故种群所在的状态空间大小是 $N * V^M$,因此种群是有限的。

$A(K+1) = T(A(K)) = T_c \circ T_m \circ T_s \circ T_{ep}(A(K))$, 其中 T_c, T_m, T_s, T_{ep} 分别表示交叉算子、变异算子、选择算子、模式交换算子的转移矩阵, 它们均与进化代数 K 无关, 因此 $A(K+1)$ 仅与 $A(K)$ 有关, 故 $\{A(n), n \geq 0\}$ 是有限齐次马尔可夫链。

设 S 为状态空间, f^* 是 S 中优化问题的最优解, 令 $A^* = \{A | \max(f(A)) = f^*, \forall A \in S\}$

定义 1 $\{A(n), n \geq 0\}$ 是种群序列, 如果对于任意的初始分布 $S_0 \in S$ 均有 $\lim_{k \rightarrow \infty} P\{A(k) \in A^* | A(0) = S_0\} = 1$, 则称算法收敛^[7]。

该定义表明, 算法收敛是指当算法迭代到足够多的次数后, 群体中包含全局最佳个体的概率接近于 1, 这种定义即为通常所说的概率 1 收敛的。

记 $P\{A(k) \in A^* | A(0) = S_0\}$ 为 P_k , 则

$$P_k = \sum_{i \in A^*} P\{A(k) = i | A(0) = S_0\}$$

记 $P\{A(k) = i | A(0) = S_0\}$ 为 $P_i(k)$, 则

$$P_k = \sum_{i \in A^*} P_i(k) \quad (1)$$

算法采用了精英保留策略(记忆抗体产生算子), 所以转移概率 $P_{ij}(k) = P\{A(k) = j | A(0) = i\}$ 有两特殊情况:

$$\text{当 } i \in A^*, j \notin A^* \text{ 时, } P_{ij}(k) = 0 \quad (2)$$

$$\text{当 } i \in A^*, j \in A^* \text{ 时, } P_{ij}(k) = 1 \quad (3)$$

定理 2 TRHEA 是概率 1 收敛的。

证明: 由式(1)得

$$P_{k+1} = \sum_{i \in A^*} \sum_{j \in A^*} P_i(k) P_{ij}(1) + \sum_{i \in A^*} \sum_{j \notin A^*} P_i(k) P_{ij}(1) \quad (4)$$

由转移概率的性质: $\sum_{j \notin A^*} P_{ij}(k) + \sum_{j \in A^*} P_{ij}(k) = 1$ 故

$$\begin{aligned} P_k &= \sum_{i \in A^*} P_i(k) = \sum_{i \in A^*} P_i(k) (\sum_{j \in A^*} P_{ij}(k) + \sum_{j \notin A^*} P_{ij}(k)) \\ &= \sum_{i \in A^*} \sum_{j \in A^*} P_i(k) P_{ij}(1) + \sum_{i \in A^*} \sum_{j \notin A^*} P_i(k) P_{ij}(1) \end{aligned}$$

由式(2) $\sum_{i \in A^*} \sum_{j \notin A^*} P_i(k) P_{ij}(k) = 0$ 故

$$P_k = \sum_{i \in A^*} \sum_{j \in A^*} P_i(k) P_{ij}(1)$$

由(4)式 $P_{k+1} = P_k + \sum_{i \in A^*} \sum_{j \in A^*} P_i(k) P_{ij}(1) > P_k$ 所以,

$$1 \geq P_{k+1} > P_k > P_{k-1} > P_{k-2} \dots$$

因此 $\lim_{k \rightarrow \infty} P_k = 1$ 。

由定义 1, TRHEA 是概率 1 收敛的, 于是定理 2 得证。

2 进化算法在智能交通系统中的应用实例

许多学者提出了改进的进化算法试图解决最短路径求解的优化问题。如文献[9]中提出了蚁群算法在 K-TSP 问题中的应用, 文献[8]则提出了一种基于模糊优化的物流配送路径(MLRP)问题研究, 并与简单遗传算法进行比较, 其效果有所提高。这里针对 ITS 的动态交通分配中多目标优化问题的特点, 提出一种求解公共交通线路换乘^[11]多目标优化问题的自适应混合进化算法(TRHEA)。文献[11]中的实际问题可以抽象如下, 目标函数:

$$\begin{aligned} \min Z &= \alpha \times C_1 T_{\text{换乘}} + \beta \times C_1 T_{\text{乘车}} + \gamma \times T_{\text{车费}} \\ T_{\text{换乘}} &= N \times C; T_{\text{乘车}} = \sum_i \sum_k (a_i^k - a_i^{k-1}); T_{\text{车费}} = \sum_i f_i(n_i) \\ \text{s. t} &\begin{cases} 0 \leq N \leq q \\ s_i^{\min} \leq d_i^k - a_i^k \leq s_i^{\max} \end{cases} \end{aligned}$$

参数说明:

$T_{\text{换乘}}$ 为乘客换乘所花费的时间; $T_{\text{车费}}$ 从起始点到终点的乘车车票花费; $T_{\text{乘车}}$ 为乘客从起始点到终点的乘车时间花费; N 为乘客从起始点到终点的换乘次数; C 为一次换乘所花费的时间。

s_i^{\min}, s_i^{\max} : i 线路上最小、最大停车时间; d_i^k : 第 i 条线从第 k 站发车的时间; a_i^k : 第 i 条线到达第 k 站的时间; $f_i(n_i)$: 第 i 条线行驶 n_i 站的费用; α : 换乘权重系数; β : 费用权重系数; γ : 时间权重系数。

c_1 系数能将出行时间转化为出行费用(元), 以保证公式 $\min Z = \alpha \times c_1 T_{\text{换乘}} + \beta \times c_1 T_{\text{乘车}} + \gamma \times T_{\text{车费}}$ 中的计算单位均为元。一般公交乘客单位时间价值为每小时人均国内生产总值^[10], 即:

$$c_1 = \frac{\text{人均国内生产总值}}{365 \times 24} \text{ (元/h)}$$

α, β, γ 还可以实时考虑个人的出行目的和爱好。

约束条件: $0 \leq N \leq q$, 我们认为在公交体系已经相当完善的大城市, 一般出行换乘 $q=2$ 次就可以到达目的地, 否则该公交网络设计不合理。

算法(TRHEA)中所需要的参数定义为: 子群组数目 $m=3$, 一般可以设定 m 为可以换乘的次数加 1, 每个子群分别存放不同换乘次数的路线。备选路径集的大小 $\max=60$, 子群体大小 $k=20$, 交叉概率 $P_c=0.9$, 变异概率 $P_t=0.01$, $P_a=0.1$, 终止代数为 25。权重系数 α, β, γ , 应该综合人们的出行目的和爱好实时考虑, 本文简化^[10]为 $\alpha=1/3, \beta=1/3, \gamma=1/3$ 。

图 3 中的实验结果是采用本文提出的 TRHEA 算法求出的站点间^[18]最优路径的迭代次数, 从图中还可以看出采用了进化算法 TRHEA 的操作算子后适应值在第 15 代时达到最优。进化算法所选路径在尽量满足最优的基础上, 还能充分考虑出行者的综合需求(通过调整权值 α, β, γ), 达到了真正意义上优化网络资源的目的。传统最短路径法虽然也能找到最优方案, 但其计算量太大, 假设含有源节点的线路有 d_1 条, 含有目的节点的线路有 d_2 条, 以一条线路上平均有 t 个节点, 换乘不超过 2 次为例, 需迭代 $d_1 \times d_2 \times t^3$ 次。若再考虑同一地铁站对应的任意两个公汽站之间可以通过地铁站换乘, 则其计算量更大。

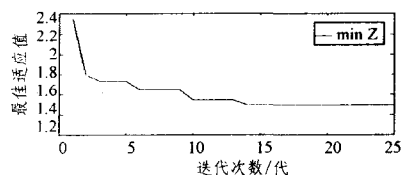


图 3 最优路径的迭代次数(TRHEA 算法)

结束语 本文针对 ITS 交通线路换乘多目标优化问题, 提出了一种求解公共交通线路换乘多目标优化问题的自适应混合进化算法(TRHEA)。TRHEA 综合多种操作算子的特性, 具有种群规模小、内在并行性及同时具备较强的全局和局部寻优能力。特别是针对交通线路换乘多目标优化问题所提出的交叉和变异操作具有可行性, 基于路径表示的染色体十进制编码方法具有创新性, 交换最优模式的协同进化使得进化过程更加稳定。我们对其收敛性进行了理论证明, 对其收敛速度进行了简要分析。仿真实验也表明, 此算法不仅能搜索到路径的最优解, 而且计算量相对于传统算法大大减少。下一步工作是充分利用自适应进化算子的特性, 研究小生境技术及协同进化机制的特点, 改进算法性能, 进一步完善智能交通网络系统的优化控制。

参考文献

[1] 刘芹, 史忠科. 混合粒子群算法求解交通路网中的车辆调度问题

- [J]. 控制与决策, 2006, 21(11): 1284-1288
- [2] Holland JH. Genetic algorithms and classifier systems: foundations and their applications[A]//Proceedings of the Second International Conference on Genetic Algorithms[C]. 1987:82-89
- [3] You XM, Liu S, Shuai DX. On parallel immune quantum evolutionary algorithm based on learning mechanism and its convergence[C]//L. Jiao, et al., eds. Proc. of ICNC06, PT1 4221. Berlin Heidelberg:Spring-Verlag, 2006: 903-912
- [4] Lydyard P M, Whelan A, Fanger M W. Instant notes in immunology[M]. Beijing: Science Press, 2001: 1-40
- [5] 游晓明, 帅典勋, 刘升. 基于免疫原理的量子进化算法及收敛性研究[J]. 控制与决策, 2007, 22(7): 749-754

- [6] 王磊, 潘进, 焦李成. 免疫算法[J]. 电子学报, 2000, 28(7): 74-78
- [7] 潘正君, 康立山, 陈毓屏. 演化计算[M]. 北京: 清华大学出版社, 1998
- [8] 张潜, 李钟慎, 胡祥培. 基于模糊优化的物流配送路径问题研究[J]. 控制与决策, 2006, 21(6): 689-692
- [9] 黄席樾, 胡小兵. 蚁群算法在 K-TSP 问题中的应用[J]. 计算机仿真, 2004, 21(12): 162-164
- [10] 杨兆升. 城市智能公共交通系统理论与方法[M]. 北京: 中国铁道出版社, 2004
- [11] 乘公交, 看奥运[EB/OL]. <http://www.shumo.com/home>

(上接第 145 页)

时间类型, 与 bookTrans 过程的输入类型一致, 并且新过程 CharterFlight 的输出参数为 boolean 类型, 也与 bookTrans 过程的输出类型一致, 因此其接口信息不会与已有过程的接口发生冲突, 可以添加该过程。

5 相关工作

目前需求演化方面的工作主要集中在需求演化的实证分析和需求演化建模两方面。需求演化的实证分析一般是指需求工程的方法和实践, 目前相关工作非常有限, 且缺乏对相关需求工程方法学之间的比较分析。这一方面是因为很难收集和析需求演化数据(软件生命周期的跨度很大); 另一方面是目前需求工程方法学缺乏对需求演化分析的支持^[8]。Hooks 和 Farry 等人认为需求质量是可以通过分析变更请求(CRs)和差异报告(DRs)来度量的, 分析结果将促使需求发生变更^[9]。Lutz 和 Mikulski 等人则认为新的需求能从操作异常中产生^[10]。

目前虽然存在很多建模方法和建模语言, 但是侧重于需求演化建模的很少。现有两种策略进行需求演化建模: 一个是使用一种建模需求的方法, 该方法可以使需求的演化更简单; 另一个是在需求模型中建模演化本身。PROTEU 工程提出了一种形式化框架用于进行需求变更的描述和推理。该形式化描述由一种目标结构框架(goal-structures framework)组成^[11]。PROTEUS 的目标结构框架在描述需求和需求之间的交互时考虑到了需求的变更。更重要的是, 该框架获取了系统和环境之间的交互, 这些交互(例如需求之间的交互和系统与环境之间的交互)形成了分析敏感度和影响的基础。Zowghi 和 Offen 提出了一种逻辑框架用于需求演化建模^[12], 它包括两个基本操作: ① 非单调推理(nonmonotonic inference), 通过领域模型来完成需求; ② 信念修正(belief revision), 通过创建新的需求来完成需求。

结束语 本文将元信息、元建模、反射和需求演化结合起来, 构造了一种在需求分析阶段支持 OWL-S 需求规约演化的反射式需求规约。反射式需求规约有助于在需求元建模框架 RGPS 的支持下, 高效率、高质量地完成需求演化建模的分析任务。我们的研究工作主要贡献在于提供了一套完整的方法对需求演化进行建模, 以一种可控、有序的方式完成需求规约的演化。

今后还需要开发一个工具, 以支持需求分析人员使用基于反射机制的需求规约演化方法。这个工具应该能够支持检索、组合、分解、分析和验证 OWL-S 需求规约及其组成元素, 支持反射机制的实现即元信息模型的抽取和需求规约演化的

完成; 还将针对交通领域的具体案例进行更深入的研究, 展示反射机制对需求规约演化的支持过程。

参考文献

- [1] Maes P. Concepts and Experiments in Computational Reflection [C]//Proceedings of OOPSLA87. ACM Sigplan Notices, New York: ACM Press, 1987: 147-155
- [2] Huang Gang, Wang Qianxiang, Mei Hong, et al. Research on Architecture-Based Reflective Middleware[J]. Journal of Software, 2003, 14(11): 1819-1826
- [3] Wang J, He K, Li B. Meta-models of Domain Modeling Framework for Networked Software[C]//Proc. of The Sixth International Conference on Grid and Cooperative Computing (GCC 2007). 2007: 878-885
- [4] Feng Zaiwen, He K. Towards Individualized Requirements Specification Evolution for Networked Software Based on Aspect[C]//International Conference of Software Process(ICSP'08). 2008: 88-99
- [5] Martin D, Ankolekar A, Burstein M, et al. OWL-S: Semantic Markup for Web Services[S]. W3C Candidate Recommendation, 2004, <http://www.daml.org/services/owl-s/>
- [6] Demers F N, Malenfant J. Reflection in logic, functional and object-oriented programming; a short comparative study[C]//Proceedings of the IJCAI95 Workshop on Reflection and Metalevel Architectures and their Applications in AI. 1995: 29-38
- [7] Costa F M. Combining meta-information management and, reflection in an architecture for configurable and reconfigurable middleware[D]. Lancaster University, 2001
- [8] Jarke M, Pohl K. Requirements engineering in 2001: (virtually) managing a changing reality[J]. Software Engineering Journal, 1994: 257-266
- [9] Hooks I F, Farry K A. Customer-Centered Products: Creating Successful Products through Smart Requirements Management [M]. Amacom, 2001
- [10] Lutz R R, Mikulski I C. Operational anomalies as a cause of safety-critical requirements evolution[J]. The Journal of Systems and Software, 2003, 65(2): 155-161
- [11] Harker S, Eason K, Dobson J. The change and evolution of requirements as a challenge to the practice of software engineering [C]//Proceedings of the IEEE International Symposium on Requirements Engineering. New York: IEEE Computer Society Press, 1999: 266-272
- [12] Zowghi D, Offen R. A logical framework for modeling and reasoning about the evolution of requirements[C]//Proceedings of the Third IEEE International Symposium on Requirements Engineering. New York: IEEE Computer Society Press, 1997: 247-257