软件人群体智能自律协调模型研究及应用

王洪泊 [1] 曾广平 [1] 王宗杰 [1] 涂序彦 [1]

摘 要 在智能自律分散系统的构建中引入面向服务的协调机制, 提出一种面向服务的软件人群体智能自律协调 (SoftMan group intelligent autonomous coordination, SMGIAC) 模型. 把服务共同体思想和全局规划有机结合起来, 考虑基于角色的服务发现/绑定策略, 构建面向服务的元组空间, 为群体协作提供统一的服务接口. 针对数字气田管网调度实际需求, 给出 SMGIAC 数学建模方法、分解协调引理及其子服务 (流) 构造算法. 实验对比结果说明, 比较现有案例匹配、专家推理两种方法, SMGIAC 总体调度准确度有较大提高, 并克服了 ReO 软件使用不便、调度方案生成时间长的缺点.

关键词 软件人, 数字气田, 群体智能, 面向服务的协调
中图分类号 TP18

SoftMan Group Intelligent Autonomous Coordination Model and Its Application

WANG Hong-Bo[1] ZENG Guang-Ping[1] WANG Zong-Jie[1] TU Xu-Yan[1]

Abstract Service-oriented coordination mechanism is adopted during the process of constructing IADS (Intelligent autonomous decentralized system) and thus an SMGIAC (SoftMan group intelligent autonomous coordination) model is proposed. Services community integrated with GP (Global planning), role-based service discovering/binding strategy is applied to constructing the service-oriented tuple space to provide an identical service interface for group cooperation. According to the current practical requirement of gas fields pipe-net scheduling, the mathematic modeling method of SMGIAC, its coordination decomposing lemma and relative concrete solvent for constructing sub-services (flows) are put forward. The experiment comparison results show that the total scheduling nicety degree of SMGIAC makes more biggish improvement than those of existing case-based matching and experts reasoning, and also SMGIAC overcomes the shortcomings of ReO such as inconvenient use, time consuming during scheduling plan of gas field pipe-net.

Key words SoftMan, digital gas field, group intelligence, service-oriented coordination

1 引言

多年来, 群体协调一直是研究者们关注的热点, 在分布式计算、软件工程等相关研究领域中已经建立了大量的协调模型, 同时也产生出多种协调语言, 用来处理实体间交互和管理活动空间, 其主要目标是有效地指导各种协调问题的建模和求解. 典型研究成果有: Carriero 和 Gelernter 提出的 Linda[1], 是一种使用最多的数据驱动协调模型, 目前 Linda 在 Java 实现上已经有了一些新的进展[2]; 在控制驱动协调模型中最有名的是 Ciancarini 和 Hankin 提出的 IWIM[3], 它通过专用的协调语言 MANIFOLD 来实现管理进程, 工作进程则可由其他程序语言稍加扩展来实现. Linda-like 协调模型的不足之处有: 1) 要求对问题有共同理解, 缺乏明确表达语义的协作; 2) 协调和计算代码不能完全分离, 增加了构建分布式系统的难度, 这样会降低组件的可重用性; 3) 协调实体的产生和消失完全是动态的, 与元组之间的对应关系不明确. 另外, 控制驱动协调模型的协调语言是独立存在的, 虽然有自己的语言要素、语法和语义, 但是, 不同的控制驱动协调模型对应不同的协调语言势必增加构建系统的难度.

数字气田管网生产调度与决策既具有代表性, 又具有特殊性, 同时也是一个需要群体协调与决策的复杂大系统[4], 我国在这方面的研究起步较晚, 近期研究成果[5, 6] 主要集中在理论建模, 尚没有功能完善、技术成熟的实用软件产品出现. 由于气田勘探、开发、生产是随时间滚动进行的, 经常有新气井或井站需并入气田管网中供气, 同样也有新用户加入进来用气. 目前国内油气田一般做法是: 在决策过程中往往还需借助于国外的相关商业软件 (例如: ReO) 模拟产生出调度方案[7], 定期对系统进行一次大改动以应对新情况, 这样就会给气田管网连续性生产带来许多不便; 传统运筹学、调度策略有其局限性[8, 9], 集中控制对此问题的解决也显得力不从心[10], 十分需要引入新的理论模型和方法.

收稿日期 2006-5-8 收修改稿日期 2006-12-7
Received May 8, 2006; in revised form December 7, 2006
国家自然科学基金 (60375038, 60374032, 60503024), 国家十五重大科技攻关项目 (2004BA616A1103), 北京市自然科学基金 (4072018) 资助
Supported by National Natural Science Foundation of China (60375038, 60374032, 60503024), National Science and Technology Tackling Key Project of the Tenth Five-year Plan of P. R. China (2004BA616A1103), Natural Science Foundation of Beijing (4072018)
1. 北京科技大学信息工程学院 北京 100083
1. Information Engineering School, University of Science and Technology Beijing, Beijing 100083
DOI: 10.1360/aas-007-0924

针对上述理论和应用中存在的问题, 本文在智能自律分散系统[11] 的构建中引入面向服务的协调机制, 提出一种面向服务的软件人[12] 群体智能自律协调模型 (SoftMan group intelligent autonomous coordination, SMGIAC).

2　SMGIAC 面向服务的体系结构

2.1　总体结构

在大系统控制和应用领域, 由于每个软件人拥有不同的资源和能力, 对于给定问题的解决方案可能超出了任何一个软件人的能力, 因此需要许多软件人共享资源, 彼此协调来解决问题[4]. 软件人群体协调是以完成一定的任务、实现总体服务为目标的. SMGIAC 在 GPGP (Generalized partial global planning) 协调机制和 TAEMS (Task analysis, environment modeling and simulation) 任务、关系的结构化描述形式[13] 中, 引入服务共同体思想, 充分发挥团队协作优势, 以求高质量完成总体服务. 总服务可以被分解为不同但又相关的子服务, 将求解任务的软件人按照子服务的关系及其所在子服务中承担的角色进行组合就形成了一个有向服务网络, 称为服务网 (Services net), 图的结点代表服务角色及其承担的任务, 存在一个根结点代表该任务组的总服务. 子服务间可能存在有：顺序关系、与关系、非确定性或关系和确定性或关系. 面向服务的 SMGIAC 总体结构如图 1 所示.

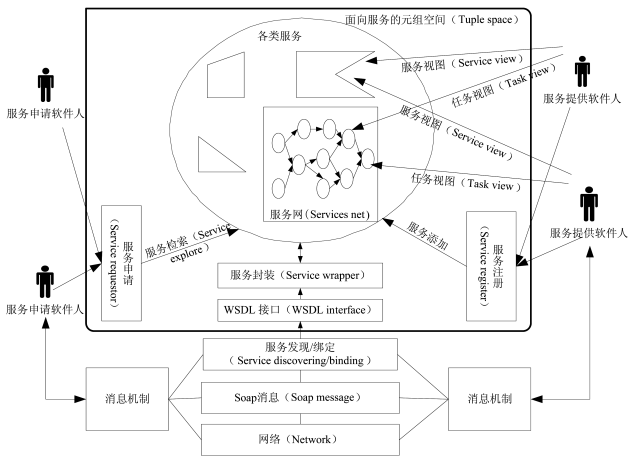

图 1　SMGIAC 的体系结构

Fig. 1　Architecture of SMGIAC

2.2　基本概念

定义 1. 服务元组 *Service-Tuple ::= (Service-Type, Service-ID, Service-Descript, Operation-Set, Ontology-DB)*. 其中, *Service-Type* 为服务类型, 同类型服务具有相同的服务接口和服务描述语义; *Service-ID* 为服务实体标识, 唯一确定 SMGIAC 计算环境中的某一服务实体; *Service-Descript* 为服务描述, 是一组服务属性对象, 其语义由服务类型决定; *Operation-Set* 是一个操作集, 是服务所要实现的操作接口; *Ontology-DB* 是服务本体库, 包括合法的属性值, 每个属性值的范围, 以及属性值之间的约束和关系, 用于定义软件人之间沟通的应用领域概念、存储资源的逻辑定义, 其目的是进行语义层次上的知识共享和重用.

定义 2. 名字服务器 *Name-Server ::= (SM-ID, Address)*. 其中, *SM-ID* 是软件人的逻辑标识的集合, 为了提供透明的通信传输, 软件人只需知道其它软件人的名字即可, 进行通信时由系统提供名字到地址的映射. 名字服务器是存储这些软件人的名字和地址的数据库, 软件人第一次运行时在名字服务器上注册相应的信息.

定义 3. 角色 *Role ::= (Role-ID, Ability, Responsibility, Access, Role-Model, Role-Relation)*. 其中, *Role-ID* 是角色的标识; *Ability* 是角色所具有的能力; *Responsibility* 表示角色所必须承担的职责; *Access* 表示角色对资源的访问权限; *Role-Model* 表示角色所属的模型, 对应于特殊应用的解决方案; *Role-Relation* 表示与其它角色之间的关系. 对每个软件人, 存在一个映射 $f : SoftMans \rightarrow Roles$. 其中, *SoftMans* 为软件人所组成的集合, *Roles* 是角色的集合. 为确保任务的实现, 在角色分配到软件人时必须满足: 至少有一个软件人扮演当前的角色, 即有约束: $\forall r \in Roles, \exists SoftMan, PlayRole\,(SoftMan, r)$.

定义 4. 交互模型 *Interact-Model ::= (Interact-ID, Purpose, Initiator, Responder, Port-pairs)*, 表示软件人之间的交互行为, 即需要与其它软件人进行协作完成任务. 由于交互可以用 π 演算的端口表示, 所以交互模型可以用一对互补的端口来表示. 其中, *Interact-lD* 表示交互协议 *ID* ; *Purpose* 表示交互的目的; *Initiator* 表示交互的发起软件人标识; *Responder* 表示交互的响应软件人标识; *Port-pairs* 表示输入、输出端口对, 如果端口设为 α, 则输入端口表示为 α, 而输出端口为 $\bar{\alpha}$.

2.3　服务封装

所有相互协调的任务按照协调模型组织在一起, 通过服务包装将协调模型各层次以 WSDL (Web services description language) 接口的形式统一对外呈现, 可以通过任务视图机制将总的任务分割给各个参与者. 服务提供方和请求方都是协调中的参与者, 都基于 Web 服务的注册、发现、绑定机制, 遵从服务描述, 以可互操作的通信协议进行通信. 参与

协作的服务、框架、活动等各种实体事实上都包装了 WSDL 格式的统一接口, 运行时系统能从中发现以各类角色参与实体中的服务描述、消息格式、端口操作与参数类型等重要信息, 并通过手工或者自动绑定到服务端口来完成相应的操作. SMGIAC 外在使用消息传递模式通信, 邮箱负责发送和接收标准格式消息, 消息处理模块负责解析消息, 消息使用 XML 进行封装.

2.4 服务元组空间

面向服务元组空间 DS 中的每个元素称作服务元组 *(Service-Tuple)*, 它是由实际字段 (Actual fields) 和规格字段 (Formal fields) 序列所组成. 服务元组的标识与选择是通过对其域值匹配 (称为模板, Template) 来完成的, 而不是通常所采用的地址选择方法. SMGIAC 以服务元组空间为核心提供进程间的同步和通信, 对服务元组提供以下三种基本操作原语. 其中假设: e 是服务元组变量, t 是模板, $System = ([P, DS]|[P \in Service\text{-}process, DS \in Service\text{-}based\text{-}Tuplespace])$.

1) 操作原语 $out(e) : [out(e).P, DS] \rightarrow [P, DS \oplus e]$;

2) 操作原语 $in(t) : \frac{\exists\, e \in DS:\ e \nabla t}{int(\vec{x}).P, DS] \rightarrow [p[e|\vec{x}], DS - e]}$;

3) 操作原语 $read(t) : \frac{\exists\, e \in DS:\ e \nabla t}{rdt(\vec{x}).P, DS] \rightarrow [p[e|\vec{x}], DS]}$.

其中, $out(e)$ 是非阻塞操作; $in(t)$ 和 $read(t)$ 是两种可能阻塞的操作. $out(e)$ 是把服务元组 e 写入 (用符号 $\oplus$ 表示) 服务元组空间; $in(t)$ 是从服务元组空间中取出 (用符号 $-$ 表示) 和 t 相匹配 (用符号 ∇ 表示) 服务元组 e; $read(t)$ 是从服务元组空间中读取服务元组. 与 $in(t)$ 的区别是 $read(t)$ 只是复制和 t 相匹配元组 e 而不是取出.

3 气田管网调度中 SMGIAC 的构建

3.1 气田管网调度的特点

在气田的实际生产中: 1) 气井的工作制度富于变化, 气井的完整生存过程为: 投产, 试产, 生产, 变更 (调增、调减、休井), 再生产, 衰竭停气, 其间气井自身工作制度变化复杂; 2) 气井的生产具有社会性, 许多口气井组成一个井站, 若干个井站 (分为采气站、集气站、输气站等) 通过物理管线联结成复杂的管网, 气井的工作制度不是单纯的个体行为, 而是需要群体实时交互与协调的社会行为; 3) 采输管网结构复杂、约束众多, 不能只从静态的拓扑结构去考查, 而是应该从动态的角度去分析和研究. 鉴于以上分析, 在软件人系统平台总体设计[14] 的基础上, 根据数字气田生产实践的需要, 考虑其有实时性、灵活性和复杂性等特点, 构建基于 SMGIAC 的数字气田管网智能调度系统.

3.2 SMGIAC 数学建模

3.2.1 可行天然气流服务网与可行服务气流

定义 5. 可行天然气流服务网可用一个五元组表示: $N = (V, A, L, U, D, T)$. 其中, V 为结点集; A 为弧集; $L, U : A \rightarrow R$ 分别为弧上的 (容量下界、上界) 权函数; $D : V \rightarrow R$ 为顶点上的权函数 (供需量); T 是时间片变量. N 上 t 时刻的一个服务流 (Service flow)x^t 是指从弧集 A 到 R 的一个函数, 即对每条弧 $(i, j) \in A$ 赋于一个实数 x_{ij}^t (称为弧 (i, j) 的流量), 如果流 x^t 满足

$$\sum_{(i,j)\in A} x_{ij}^t - \sum_{(j,i)\in A} x_{ji}^t = d_i^t, \quad \forall\, i \in V \tag{1}$$

$$l_{ij} \le x_{ij}^t \le u_{ij}, \quad \forall\, (i, j) \in A \tag{2}$$

则称 x^t 为可行服务流 (Feasible service flow) . 其中式 (1) 和式 (2) 分别是流量守恒和容量约束条件. 至少存在一个可行服务流的网称为可行服务流网络 (Feasible service flow network). 可见: 1) 当 $d_i^t > 0$ 时, 表示在 t 时刻有 d_i^t 个单位的流量从网络外部流入该结点, 因此在 t 时刻顶点 i 称为供应点 (Supply node) 或源 (Source) 结点, 由许多源结点构成的集合记为: $Sp = (i|d_i^t > 0)$; 2) 当 $d_i^t < 0$ 时, 表示在 t 时刻有 d_i^t 个单位的流量从该结点流出到网络外部 (或被网络吸收), 因此在 t 时刻顶点 i 称为需求点 (Demand node) 或汇 (Sink) 结点, 由许多汇结点构成的集合记为: $Si = (i|d_i^t < 0)$; 3) 当 $d_i^t = 0$ 时, 在 t 时刻顶点 i 称为转运点 (Transshipment node) 或平衡点, 由许多转运结点构成的集合记为: $Tr = (i|d_i^t = 0)$. 此外, 对于可行服务流网络, 必有: $\sum_{(i)\in V} d_i^t = 0$, 即所有结点上的供需量之和为零是网中存在可行气流的必要条件.

3.2.2 路流服务与环流服务

定义 6. 可行服务流网络中的流 $x^t : A \rightarrow R$, 其中, 如果 N 中存在一条从源 $i_0 \in Sp$ 到汇 $j_0 \in Si$ 结点的有向路径 $Path(i_0, j_0)^t$, 使得

$$0 \le v = x_{ij}^t \le u_{ij}, \quad \forall\, (i, j) \in Path(i_0, j_0)^t \tag{3}$$

$$x_{ij}^t = 0, \quad \forall\, (i, j) \in A \diagdown Path(i_0, j_0)^t \tag{4}$$

则称 x^t 为路流服务 (Path flow service), 称 v 为该路流的流值 (流量), 记作 $x_{Path(i_0,j_0)}^t(v)$. 当 $v = 0$ 时, 称该路流为零路流服务, 否则称为非零路流服务.

同样, 如 N 中存在一个有向环 $Cycle$ (环中不含源或汇结点), $\forall\, i_1 \in Tr$ 使得

$$0 \le v = x_{ij}^t \le u_{ij},\ \forall\, (i, j) \in Cycle(i_1, i_i)^t \tag{5}$$

$$0 \le v = x_{ij}^t \le u_{ij},\ \forall\, (i, j) \in A \diagdown Cycle(i_1, i_i)^t \tag{6}$$

则称 x^t 为环流服务 (Cycle flow service), 称 v 为该环气流的流值 (流量), 记作 $x^t_{Cycle(i_1,i_i)}(v)$. 当 $v=0$ 时, 称为零环流服务, 否则称为非零环流服务.

定义 7. 路流服务承输因子

$$Sp_{rate}(x^t_{Path(i_0,j_0)}(v))=\frac{x^t_{Path(i_0,j_0)}(v)}{x^t(v)} \tag{7}$$

定义 8. 环流服务承输因子

$$Sc_{rate}(x^t_{Cycle(i_1,i_1)}(v))=\frac{x^t_{Cycle(i_1,i_1)}(v)}{x^t(v)} \tag{8}$$

定义 9. 源结点承输因子

$$Source_{rate}(i)=\frac{\sum\limits_{i_0=i}x^t_{Path(i_0,j_0)}(v)}{x^t(v)} \tag{9}$$

定义 10. 汇结点承输因子

$$Sink_{rate}(i)=\frac{\sum\limits_{j_0=j}x^t_{Cycle(i_0,j_0)}(v)}{x^t(v)} \tag{10}$$

3.3 SMGIAC 分解协调引理

引理 1. 在可行天然气流服务网中, t 时刻任一可行服务流对应的服务元组空间 DS 一定可划分为若干个路径集 (Path set), 流值 $x^t(v)$ 可知的 (稳定) 路流服务元组 (Path-service-tuple) 和若干个 (稳定) 环流服务元组 (Cycle-service-tuple), 即有

$$x^t(v)=\sum_{i_0\in Sp}^{j_0\in Si}x^t_{Path(i_0,j_0)}(v)+\sum_{i_i\in Tr}x^t_{Cycle(i_1,i_i)}(v) \tag{11}$$

并有, 非零路流服务元组对应的有向路径从一个 (或多个) 源点指向一个 (或多个) 汇点; 至多有 $m+n$ 个路流服务元组和环流服务元组为非零流, 且其中至多有 m 个流环服务元组为非零流, 其中 m 是 N 中的结点数目, n 为 N 中边的数目.

证明. 记可行流为 x^t , 设 i_0 是可行服务流网中的一个源点, 则存在弧 $(i_0,i_1)\in A$ 使得 $x_{i_0,i_1}>0$. 如果 i_1 是网络中的一个汇点, 则找到一条从一个源点指向一个汇点的有向路径; 否则, 从 i_1 出发, 重复上述过程, 直到找到一个汇点或再次遇到前面已经经过的某个顶点时为止, 即直到下列两种情况之一出现为止:

情况 1. 找到一条从一个源点 i_0 指向一个汇点 i_k 的有向路径 $Path$. 此时, 定义

$$x^t(Path)=\min(d_i,-d_i,\min(x^t_{i,j}|(i,j)\in Path)) \tag{12}$$

如果 $\frac{\partial Sp_{rate}(x^t_{Path(i_0,i_k)}(v))}{\partial t}\approx 0$, 则构造了一个流值为 $v^t(Path)$, 路径集为 $Path$ 的非零 (稳定) 路流服务元组 ($Path-service-tuple$). 将其写入 DS 中: $[out(Path\text{-}service\text{-}tuple\ (v^t(Path))).P,DS]\rightarrow[P,DS\oplus\ Path\text{-}service\text{-}tuple(v^t(Path))]$. 此时, 在网中对 $\forall(i,j)\in\ Path$ 重新定义

$$d_{i_0}=d_{i_0}-v^t(Path) \tag{13}$$

$$d_{i_k}=d_{i_k}+v^t(Path) \tag{14}$$

$$x_{i,j}=x_{i,j}-v^t(Path) \tag{15}$$

否则, 从源点 i_0 重新开始; 如果 i_0 遍历完毕, 则重新从另一源点 $i_0^{'}$ 开始, 即 $i_0=i_0^{'}$.

情况 2. 找到一个有向环 $Cycle$. 此时, 定义

$$v^t(Cycle)=\min(x_{ij}|\forall(i,j)\in Cycle) \tag{16}$$

如果 $\frac{\partial Sc_{rate}(x^t_{Cycle(i_1,i_1)}(v))}{\partial t}\approx 0$, 则构造了一个流值为 $v^t(Cycle)$, 路径集为 $Cycle$ 的非零 (稳定) 路流服务元组 ($Cycle-service-tuple$). 将其写入 DS 中: $[out(Cycle\text{-}service\text{-}tuple\ (v^t(Cycle))).P,DS]\rightarrow[P,DS\oplus\ Cycle\text{-}service\text{-}tuple(v^t(Cycle))]$. 此时, 在网中对 $\forall(i,j)\in Cycle$ 重新定义

$$x_{i,j}=x_{i,j}-v^t(Cycle).$$

否则, 从源点 i_0 重新开始; 如果 i_0 遍历完毕, 则重新从另一源点 $i_0^{'}$ 开始, 即 $i_0=i_0^{'}$.

对重新定义的网, 重复上述过程, 直到所有顶点变成平衡点. 然后, 对重新定义的网, 找到一个至少有一条出弧上的流量为正的顶点, 继续重复上述过程, 这时只有情况 2 会出现, 即一定获得一个非零 (稳定) 环流服务元组. 重复上述过程, 直到定义的流 x 成为零流为止. 最后, 可按照汇点对路流服务元组进行合并. □

上述引理是将 Ford 和 Fulkerson 的流分解算法[15] 推广到多源、多汇的情况下, 并加入了时间片 t 因素对流服务稳定性约束, 这在实际应用中显得非常重要. 例如: 在气田采输管网调度过程中, 保证天然气的平稳采输是一个重要的技术指标. 在正常情况下, 如果管网中总有频繁的大流量变更, 可能会增加生产成本, 引起流量控制类设备的损坏, 严重的会造成生产秩序的混乱, 带来不必要的经济损失.

4 算法实例

根据上节引理的构造性证明, 得到一种 SMGIAC 的分解协调算法, 下面结合实例进行阐述.

4.1 问题

图 2 是某气田集输管网示意图. 在 t 时刻, 气源 Source 1, Source 2, Source 3 的采气工作制度分别为 $20, 8, 6$; 两个用户 Sink 1, Sink 2 的用气工作制度分别为 $17, 14$; Tran-node $1, \cdots, 6$ 是六个集输站点; 相关管线分布如图 2 所示. 对此, 如何合理安排调度方案, 才能保证气田管网的安全经济运营?

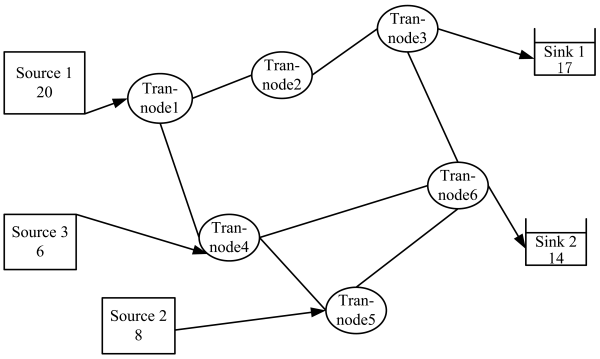

图 2 气田管网示意图 (气量单位: 10^6/d)

Fig. 2 Model of gas fields pipe-net (unit: 10^6/d)

4.2 算法描述

对非零路气流服务来说: 气井或井站 (网中的源点) 是服务提供方, 而天然气用户 (网中汇点) 是服务申请方, 集输站场或管线等是服务的约束条件或中介; 而非零环气流服务是一种特殊的自耗式服务. 算法在实际操作中, 要遵循大者优先原则, 其具体解释为: 1) 先高压管线气井组, 后低压管线气井组; 2) 产气量大的气井所在进气点优先, 按产气量对进气点建立一个源点集合: $InitSet\,(GasStation, Sp)$; 3) 输气量大的管线所在管道优先, 按输气量对管线建立一个管线容量约束矩阵 $InitNetFlux\,(vedge[i][j])$; 4) 用气量大的用户所在输配站点优先, 按用气量对用户 (或输配站) 建立汇点集合 $InitSet\,(GasUser, Si)$; 输配站集合 $InitSet\,(Trannode, Tr)$.

SMGIAC 路气流服务与环气流服务的物理意义是: 1) 所有的非零环流之和就对应于整个管网的 (实时) 动态储气量 DQ; 2) 源点对所经路径求和即是其对应 (井站) 最优合理产气量 P; 3) 汇点对所经路径求和即是其对应 (用户) 的最佳供气量 Q.

4.3 运行结果

如图 3 所示, SMGIAC 将整个气田管网气流分解为两个路气流服务和一个环气流服务. 其中 Sink 3 为一个虚拟汇点, 表示相应环气流服务对天然气的消耗, 即: 管网自耗气流量为 3 (10^6 /d). 各路气流服务、环气流服务和结点的承输因子, 分别如表 1 和表 2 所示.

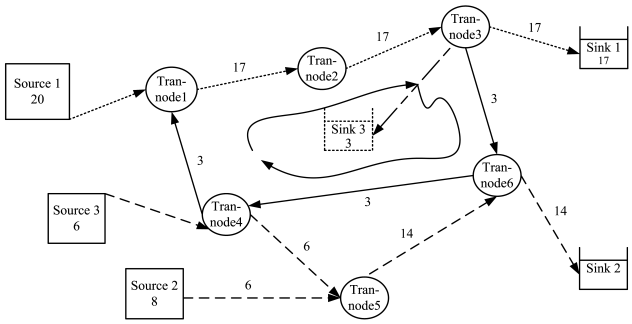

图 3 基于 SMGIAC 的调度结果 (气量单位: 10^6/d)

Fig. 3 Schedule result based on SMGIAC (unit: 10^6/d)

表 1 服务的流量及承输因子

Table 1 Flux and rate of service (unit: 10^6/d)

服务流 (Service flows)	流量	Sp_{rate}	Sc_{rate}
Source 1 → Sink 1	17	50 %	/
Source 2,3 → Sink 2	14	41.12 %	/
Tran-node 1 → Tran-node 1	3	/	8.88 %

表 2 结点流量及承输因子

Table 2 Flux and rate of nodes (unit: 10^6/d)

结点	流量	So_{rate}	Si_{rate}	$Tran_{rate}$
Source 1	20	58.82 %	/	/
Source 2	8	23.53 %	/	/
Source 3	6	17.65 %	/	/
Sink 1	17	/	50 %	/
Sink 2	14	/	41.18 %	/
Sink 3	3	/	8.82 %	/
Tran-node 1	20	/	/	58.82 %
Tran-node 2	20	/	/	58.82 %
Tran-node 3	20	/	/	58.82 %
Tran-node 4	9	/	/	26.47 %
Tran-node 5	14	/	/	41.18 %
Tran-node 6	17	/	/	50 %

5 效果分析

5.1 与案例匹配、专家推理的比较

某日 Source 3 停气检修, 管网总进气量减少, 根据天然气调度管理应急工作流程, 运用 SMGIAC 与 DSS (决策支持系统)[8] 进行案例匹配、专家推理两种方法的实验数据对比分析结果, 如表 3 所示 (注: 表中的负数表示供气量 < 需气量).

案例匹配 (匹配成功的情况下, 调度总误差绝对值为 27.71): 根据故障地点、故障类型、最近时间、事故处理周期等匹配条件, 选用距离最近的一天 Source 3 发生故障这一案例, 通过人机对话, 对重要结点的参数根据实际情况调整后进行了匹配. 其特点是需要大量的多种工况下历史案例, 适合于常规事故的处理, 对于突发事故, 难以匹配成功.

专家推理 (调度总误差为 58.7): 根据人机对话, 选取气源事故 (Source 3) 停气检修这条专家知识, 对所涉及的需要调整的气源、用户可根据实际生产情况重新进行优先级别的排列或改变相关条件值进行推理. 其特点是精度高, 但需要大量的人工参与, 不适合于应急调度. SMGIAC 的调度总误差为 9.5, 比起案例匹配、专家推理两种方法, 总体调度准确度有较大的提高. SMGIAC 无需大量历史案例, 也可减轻专家推理过程中大量人工参与, 通过相关承输因子的计算, 能对管网整体采输气量影响的大小做出判断, 可方便识别出管网中的关键服务 (流), 从而加以重点监护, 同时克服了案例匹配的不足, 为调度专家科学决策提供可视化依据. 此外, SMGIAC 算法本身就是从子服务 (流) 协调动态产生的角度设计的, 所以也便于气井工作制度的制定和生产调度指令的快速下达.

表 3　案例匹配, 专家推理与 SMGIAC 对比

Table 3　Result comparison of case-based reasoning, expert reasoning, and SMGIAC (unit: 10^6/d)

结点	应急流量	CBR 误差	ER 误差	SMGIAC 误差
Source 1	196.9	3.1	3.1	1.8
Source 2	91.1	14.83	15.9	1.5
Source 3	/	/	/	/
Source 4	344.2	-7.16	53.8	6.7
Source 5	397.2	-7.2	2.8	-1.9
Source 6	79	6.84	21	13.5
Source 7	67	-5.7	14.4	3.2
小计	1175.4	4.71	111	24.8
Sink 1	160.5	22	19.5	17.5
Sink 2	129.3	-7	0.6	0.8
Sink 3	149.2	-17.94	-14.5	-3.3
Sink 4	164.3	24.83	31	-5.7
Sink 5	83.2	17.23	14.1	7
Sink 6	87.3	-6.7	1.6	-1
小计	773.8	32.42	52.3	15.3
合计	401.6	-27.71	58.7	9.5

5.2　与 ReO 的比较

SMGIAC 与 ReO 仿真结果对比表明 (如图 4 所示), 两者数据相差不大, 压力差值在 0 ∼ 0.4 Mpa, 结点流量的吻合情况更好, 说明 SMGIAC 生成的数字气田管网应急调度方案是可信的, 能够指导气田的生产.

值得注意的是, SMGIAC 克服了 ReO 的一些不足: ReO 是离线仿真工作方式, 参数设置多, 只有少数高级专业人员才会使用, 并且生成调度方案时间周期长, 很难适用于气田管网实时生产调度; 而 SMGIAC 是在线工作模式, 界面友好, 普通工作人员就会使用, 可实时生成调度方案, 非常适用于气田管网的应急调度.

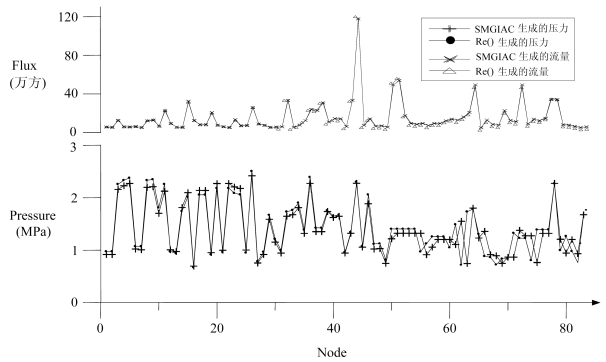

图 4　SMGIAC 与 ReO 的仿真对比结果

Fig. 4　Simulating comparison result of SMGIAC and ReO

6　结论

软件人 (SoftMan) 是物理机器人在网络世界中的虚拟化, 比通常的智体 (Agent) 更富有人工生命的活性和功能, 特别是拟人的活性和功能. 文献 [16] 更为详细地分析了软件人与智体的区别与联系, 限于篇幅, 这里不再赘述.

服务是人类社会特有的一种公共协调体系, 受此启发, 论文提出面向服务的软件人群体智能自律协调模型 SMGIAC. 与已有的多智体协调机制相比, SMGIAC 的特点体现在以下三个方面：1) 将传统的单机计算或分布计算任务转化为由实现各种服务的各类软件人共同承担的网络计算, 从而能充分发挥群体协调的优势, 对复杂问题的解决和任务的完成提供了一个良好的支撑环境; 2) 可以把一个大系统分解为几个子服务 (流) 分别建模; 鉴于这种协调分解是在线进行的, 所以子服务 (协作群体) 的产生是动态的, 这样有利于提升系统的自适应能力; 3) SMGIAC 是基于面向服务计算思想设计的, 非常便于与 Web 具体应用结合起来, 用于解决复杂网络背景下服务特征明显的一大类实际问题.

致谢

向对本文的工作给予支持和建议的国家十五数字气田项目研究与开发组的全体同仁, 以及中石化西南分公司采输处的有关领导和专家表示衷心的感谢.

References

1 Carriero N, Gelernter D. *How to Write Parallel Programs: A First Course*. Mass: MIT Press, 1992. 19∼32

2 Wells G C. New and improved: linda in java. *Science of Computer Programming*, 2005, **59**(8): 82∼96

3 Ciancarini P, Hankin C. The IWIM model for coordination of concurrent activities. In: Proceedings of the First International Conference on Coordination Model. Berlin, Germany, 1996. 34∼56

4 Tu Xu-Yan, Wang Cong, Guo Yan-Hui. *Large System Cybernetics*. Beijing: Beijing University of Posts and Telecommunications Publishing House, 2005
(涂序彦, 王枞, 郭燕慧. 大系统控制论. 北京: 北京邮电大学出版社, 2005)

5 Yang Yi, Li Chang-Jun, Shang Shu-Ya. Study on gas transmission and distribution optimization of pipeline system. *Natural Gas Industry*, 2006, **26**(1): 123∼125
(杨毅, 李长俊, 尚蜀娅. 天然气管网输配气量优化研究. 天然气工业, 2006, **26**(1): 123∼125)

6 Bai Yi. Research on Chuanxi Natural Gas Pipeline System Optimization and Simulation [Ph. D. dissertation], Southwest Petroleum University, 2004
(柏毅. 川西地区天然气管网优化及仿真技术研究 [博士学位论文], 西南石油学院, 2004)

7 Zhang Bai-Ling, Wang Yu-Sheng. The new field spirit farmland loses the tube at gather the net system excellent turn to imitate a research. *Natural Gas Industry*, 2002, **22**(6): 119∼121
(张百灵, 王雨生. 新场气田集输管网系统优化模拟研究. 天然气工业, 2002, **22**(6): 119∼121)

8 Zhu Yong, Luo Min, Liu Wei. DSS application in urgent gas dispatching. *Natural Gas Industry*, 2004, **24**(11): 131∼136
(朱勇, 罗敏, 刘巍. DSS 技术在天然气应急调度管理中的应用. 天然气工业, 2004, **24**(11): 131∼136)

9 Uraikul V, Chan C W, Tontiwachwuthikul P. MILP model for compressor selection in natural gas pipeline operations. *Environmental Informatics Archives*, 2003, **1**(1): 138∼145

10 Rios-Mercado R Z, Kim S, Boyd E A. Efficient operation of natural gas transmission systems: a network-based heuristic for cyclic structures. *Computers and Operations Research*, 2006, **33**(8): 2323∼2351

11 Tu X Y, Tang T. Intelligent autonomous decentralized system. In: Proceedings of IEEE International Workshop on Autonomous Decentralized System. IEEE, 2002. 10∼15

12 Zeng Guang-Ping, Tu Xu-Yan. SoftMan. In: Proceedings of the l0th China Aossoaciton Artificial Intelligence National Conference. Beijing: Beijing University of Posts and Telecommunications Publishing House, 2003. 567∼572
(曾广平, 涂序彦. 软件人. 中国人工智能学会全国第十届学术年会论文集. 北京: 北京邮电大学出版社, 2003. 677∼682)

13 Decker K. Environment Centered Analysis and Design of Coordination Mechanisms [Ph. D. dissertation], University of Massachusetts, 1995. 121∼150

14 Wang Hong-Bo, Ban Xiao-Juan, Zeng Guang-Ping, Yin Yi-Xin, Tu Xu-Yan. Virtual robot in network environment-softman system platform design. *Computer Science*, 2005, **32**(8): 152∼154
(王洪泊, 班晓娟, 曾广平, 尹怡欣, 涂序彦. 网络环境下虚拟机器人-SoftMan 系统平台总体设计. 计算机学, 2005, **32**(8): 152∼154)

15 Ford L R, Fulkerson D R. *Flows in Networks*. USA: Princeton University Press, 1962

16 Lu Q L, Zeng G P, Zhang W, Tu X Y. Softman and agent. In: Proceedings of IEEE International Conference on Networking, Sensing and Control. IEEE, 2005. 904∼907

王洪泊 北京科技大学信息工程学院讲师. 2007 年获得北京科技大学博士学位. 主要研究方向为人工智能与人工生命. 本文通信作者.
E-mail: foreverwhb@126.com
(**WANG Hong-Bo** Lecturer at Information Engineering School, University of Science and Technology Beijing. He received his Ph. D. degree from University of Science and Technology Beijing in 2007. His research interest covers artificial intelligence and artificial life. Corresponding author of this paper.)

曾广平 北京科技大学信息工程学院教授. 主要研究方向为分布/迁移/网络计算, Linux 操作系统构造与嵌入式系统, 智能网络与智能通信.
E-mail: zgping20012002@yahoo.com.cn
(**ZENG Guang-Ping** Professor at Information Engineering School, University of Science and Technology Beijing. His research interest covers distributed-migrating and networking computation, Linux architecture construction and embedded system, and intelligent network and intelligent communication.)

王宗杰 北京科技大学信息工程学院讲师. 主要研究方向为人工智能及应用.
E-mail: wangzongjie@vip.sina.com
(**WANG Zong-Jie** Lecturer at Information Engineering School, University of Science and Technology Beijing. His research interest covers artificial intelligence and its application.)

涂序彦 北京科技大学信息工程学院教授. 主要研究方向为人工智能, 智能控制、智能管理, 大系统控制论, 生物控制论, 人工生命.
E-mail: tuxuyan@126.com
(**TU Xu-Yan** Professor at Information Engineering School, University of Science and Technology Beijing. His research interest covers artificial intelligence, intelligence control, intelligent management, large system cybernetics, biology cybernetics, and artificial life.)