

# 研究使用 GPT 构建大语言模型智能母基金决策投资支持系统

朱铮雄 周智宏

(中海服信息科技股份有限公司 上海 200010)

**摘要** 构建基于 GPT 的智能大语言模型基金决策支持系统,以有效应对金融市场的复杂性。通过大语言模型技术,系统提供深层次的基金和项目洞察,同时进行风险计量。采用 Python 与 Flask 构建,整合了 OpenAI GPT 模型,通过嵌入式 AI 模式提升了投前管理、初筛、评分和风险评估的效能,旨在提高整体决策效率。未来的改进方向包括对模型的优化和系统的扩展。这一创新将推动金融领域数字化转型,进一步提升投资效率。

**关键词** 人工智能大语言模型 GPT Transformer OpenAI Python Flask 母基金

**中图分类号** TP3 **文献标志码** A **DOI**:10.3969/j.issn.1000-386x.2024.05.004

## BUILDING A LARGE LANGUAGE MODELING INTELLIGENT FUND OF FUNDS DECISION MAKING AND INVESTMENT SUPPORT SYSTEM USING GPT

Zhu Zhengxiong Zhou Zhihong

(Zhong Hai Fu Information Technology Co., Ltd., Shanghai 200010, China)

**Abstract** The paper aims to construct an intelligent large language model fund decision support system based on GPT to effectively deal with the complexity of financial markets. Through the large language modeling technology, the system provided deep fund and project insights along with risk measurement. Built with Python and Flask, it integrated OpenAI GPT models and enhanced the efficacy of pre-investment management, initial screening, scoring and risk assessment through embedded AI models, so as to improve overall decision-making efficiency. Future improvements included optimization of the model and expansion of the system. This innovation would drive digital transformation in the financial sector and further improve investment efficiency.

**Keywords** Artificial intelligence large language modeling GPT Transformer OpenAI Python Flask Fund of funds

## 0 引言

在当前金融市场的高度复杂性和信息爆炸的环境中,基金管理行业面临着越来越多的投资挑战。私募股权母基金(Fund of Fund)是指以私募股权投资基金为主要投资标的,对私募股权基金投资的项目目标的公司进行间接投资<sup>[1]</sup>,母基金的管理者需要面对市场波动、信息过载、资产配置等多方面的复杂情境。在这样的环境下,引入先进的人工智能技术成为提高母基金投资管理效能的迫切需求。

本文旨在构建一套基于 GPT(Generative Pre-trained

Transformer)的大语言模型智能基金决策支持系统,通过 Embedding 模式,提升原有的母基金系统的投资能力。系统充分利用 GPT 的语言理解和生成能力,为基金经理提供更深层次的基金、项目、市场洞察和决策建议,帮助其更好地应对市场波动、优化资产配置,并提高基金投资效率,提升整体回报率。

大语言模型智能母基金投资决策支持系统的建设对于提高投资决策的科学性和效率具有重要意义。通过引入近期火爆全球的、世界上最先进的 OpenAI 的 GPT 平台。我们有望提供更全面的信息处理和更准确的决策辅助,从而为投资团队提供更有竞争力的投资建议。这将促进母基金投资领域的数字化转型,提高

投资决策效率。

## 1 相关技术

### 1.1 Transformers 模型概述

Transformers 模型是一种基于自注意力机制 (self-attention) 的深度学习模型,最初提出用于自然语言处理任务,如机器翻译。其基本框架为六个编码模块组成的编码器<sup>[2]</sup>,编码器中每个模块由自注意力机制及前馈神经网络组成,使模型能够在处理序列数据时同时关注输入序列的所有位置,从而捕捉长距离依赖关系。这种架构相较于传统的循环神经网络 (RNN) 和长短时记忆网络 (LSTM) 更容易并行化,因而在训练效率上具有显著优势。

### 1.2 GPT 概述

GPT (Generative Pre-Training Transformer), 是一种基于 Transformer 构架的语言模型<sup>[3]</sup>。

首先,利用 Transformer 的自注意力机制,GPT 能够捕捉文本中的长距离依赖关系,更好地理解语境和上下文关系。在金融领域,特别是对于需要考虑文本上下文的任务,如分析财务报告或投资协议,GPT 能深入理解,展现出卓越性能。

其次,金融文本通常包含大量专业术语、复杂句式和结构,而 GPT 的 Transformer 结构使其能够更有效地处理这些复杂的文本结构,充分体现了金融领域信息的抽取和分析能力。

再次,GPT 作为生成式模型,具备生成新文本的能力,为深入挖掘和理解金融数据提供了有力支持。

在母基金投资决策中,需要进行分析投资报告、数据尽调、风控预警、生成报告、趋势分析等多项任务,GPT 的特性使其成为基金管理领域处理金融文本和决策生成的理想工具。

### 1.3 OpenAI 技术平台概述

OpenAI 成立于 2015 年,致力于研究通用人工智能 (AGI) 发展,平台发展 8 年,估值超 300 亿美元,预计 2024 年收入超 10 亿美元<sup>[4]</sup>。该机构开发的 GPT 具备卓越的语义理解和生成能力,为解决自然语言处理任务提供了强大的工具。作为全球最先进的人工智能大模型机构,OpenAI 引领了全球人工智能大模型的发展方向,是国内众多大模型机构如文心一言、智谱 AI 等所借鉴的典范和衡量的标准。我们选择 OpenAI 的 GPT 模型作为研究智能母基金投资决策支持系统的基座模型,主要考虑实际运行效果和通用性,为将来

能够使用国产大模型以及自训练大模型,打下坚实基础。

OpenAI API 提供了对 GPT3.5/GPT4 模型的访问。通过 API 请求,开发者可以向 GPT3.5/4 模型提供输入文本并获取生成的输出。

### 1.4 Python 与轻量级 Web 框架 Flask

Python 被广泛应用于数据处理、科学计算 (如 NumPy、Pandas、Scikit-learn) 和人工智能、深度学习框架 (如 TensorFlow 和 PyTorch) 等大规模神经网络的开发<sup>[5]</sup>。Flask<sup>[6]</sup> 是基于 Python 的轻量级 Web 应用框架,适合快速构建微服务,能在最小侵入的情况下为决策支持系统提供 RestfulAPI 智能服务。

## 2 需求分析

### 2.1 现有业务系统主要问题

现有母基金投资管理系统上线于 2020 年,采用 SpringBoot + Vue.js 构建,涵盖管理人、基金、募资、投前、投资、投后等模块。随着 FOF 股权市场规模不断扩展,接洽待处理的项目几何级增长,现有母基金投资管理系统面临以下问题:

投资流程冗长:随着项目数量规模扩大,系统设计的流程显得冗长,操作繁琐,导致了流程堆积的现象,用户难以及时完成相关流程。

信息离散困境:大量的立项材料分散在投资经理的本地电脑上,系统仅剩空壳,形式主义明显,无法充分发挥信息管理的效益。

数据处理滞后:在研究项目公开市场数据时,系统的手动录入方式导致了信息处理滞后,项目数据无法及时更新。

尽调形同虚设:风控经理对项目立项和尽调材料的完整性预检查存在不足,缺乏有效手段来预防潜在风险。

评价体系混乱:现有系统的评价体系缺乏客观评分标准,导致对管理人和项目的准确评估受到影响。

绩效反馈不足:绩效评分结果的反馈不够具体,投资决策委员会难以深入了解每个评价维度的贡献度和改进空间,制约了决策的科学性。

数据归因出错:缺乏有效数据,原有系统的数据模型无法充分驱动,无法进行有力的数据归因。

这些问题共同影响了系统运行的效率、实用性和数据准确性,需要对系统全面优化和改进。

### 2.2 业务流程改进设计

在原有系统中的投前管理、投资管理、投后管理模块中结合 GPT 大模型的特点,来进行设计决策支持系统,支持系统通过以下几个方面来解决原有系统的不足。

语义理解:GPT 通过深度学习离散文档,自动扫描和理解,准确捕捉关键信息,为项目提供更全面的初步评估。

自动化操作:通过 GPT 的 function ai 功能,将 ai 嵌入进系统,替换掉大量人工搜索、手动录入等繁琐功能。

数据初筛与尽调:结合公开市场数据,购买的专业数据,根据预设的校验模型,确保项目数据的准确性和及时性,并给于尽调初筛有效支持。

提示重点研究:GPT 分析已有数据,结合预设的经验模型,并提示下一步项目研究的重点,引导投资经理更有针对性地进行深入研究,反馈进系统。

## 3 系统设计与实现

### 3.1 系统设计

为了降低对原有系统影响,使用 Flask 构建 Restful api 嵌入原有系统进行交互,最大程度降低对原有组件影响,保证系统正常运行,系统结构设计如图 1 所示。

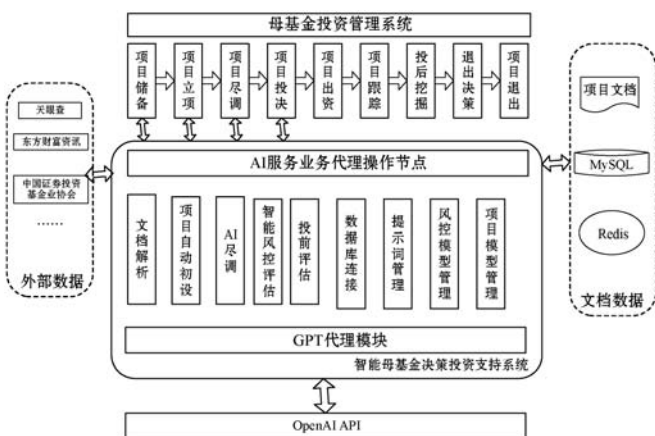


图 1 系统构架

为了增强 AI 的决策辅助功能,我们采用嵌入式 AI 构架模式。嵌入式 AI 构架通常是指将模型嵌入到其他系统或应用程序中,以提供特定的功能或增强现有功能。在这种模式下,AI 的作用相当于执行命令的工具,而原系统用户继续担任决策者和指挥者的角色<sup>[7]</sup>。我们通过 Flask 调用 OpenAI API Function Calling 库,远程应用 GPT 大模型进行分析计算,并回调本地函数<sup>[8]</sup>,以完成增强的智能功能。其调用流程如图 2 所示。

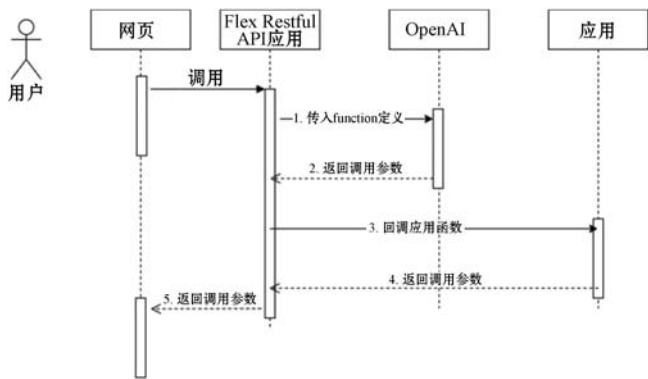


图 2 系统调用流程

### 3.2 数据模型设计

在现有的数据设计基础上,增加初筛设计表,此表将记录每一次上传初筛报告内容及修改报告的内容,作为评价依据。

增加风险报告数据表,通过 GPT 扫描初筛报告和相关项目表信息,记录 GPT 分析风险以及风控人员对于报告的分析。

增加评价数据表,通过调用 GPT 模型库,记录项目进行评分得出的结果。

各表关系如图 3 所示。

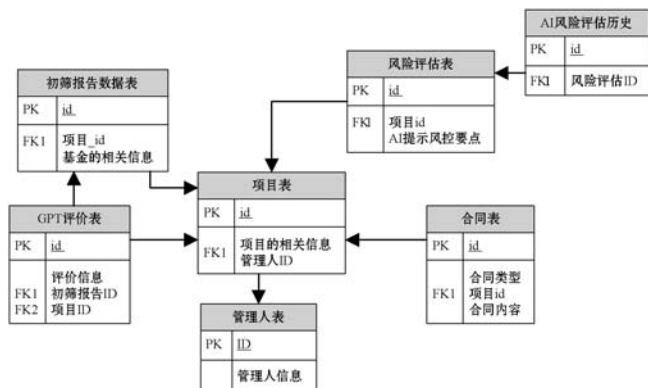


图 3 数据表结构

### 3.3 提示词(Prompt)模型设计

提示工程是指构建可以被大模型理解和解释的文本结构。有效的提示工程不仅能提高大模型的准确性和效率,还能帮助模型更好地理解 and 处理复杂的金融数据和环境。

提示词 = 角色 + 指示 + 背景 + 属性 + 模型要求 + 输入文本 + 输出结构

角色:对系统角色描述,如你是一个股权投资经理。

指示:对任务进行描述,如读取文档,仔细分析最近 3 年财务报告。

背景:添加针对特定背景业务(如银行、高科技、医药等)的背景信息。

属性:设置金融特有的属性,如从合格机构投资者

条件判断等专业术语。

**模型要求:**根据我们的经验总结数据模型,如投资策略得分,按照母基金的投资策略与管理人投资策略匹配性(50分)、行业发展情况(30分)、分散风险情况(20分)进行评分。

**输入文本:**定义金融数据或特定业务参数作为输入。如输入一篇私募基金管理人的访谈纪要等。

**输出结构:**输出的格式描述,以便后续系统模块能直接处理输出结果,比如输出结果为 json 格式,其中 3yr\_portfolio 字段取值为 List 类型,取值必须为最近投资组合的清单。

相关提示词会存储在数据库中,可以不断地优化调整,方便系统使用时候动态调用。

### 3.4 基本组件接口的设计与实现

#### 3.4.1 基本工具类,完成 GPT 调用

考虑系统对 GPT 模型的可移植性,我们封装了 OpenAI 客户端代码,设计了 2 个主要基础函数,分别是直接调用 OpenAI 函数和 OpenAI 结合本地函数。

(1) 最基本 GPT 调用函数,获取 GPT 文本反馈。

# 定义一个 client,用来调用 OpenAI

```
client = OpenAI(
    api_key = os.getenv("OPENAI_API_KEY"),
    base_url = os.getenv("OPENAI_API_BASE")
)

# 定义一个 GPT 调用函数,获取反馈
def get_gpt_call(prompt, model = "gpt-3.5"):
    messages = [{"role": "user", "content": prompt}]
    response = client.chat.completions.create(
        model = model,
        messages = messages,
        temperature = 0,
        #模型输出的随机性,0 表示随机性最小
    )
    return response.choices[0].message.content
```

(2) 调用 OpenAI 并反调本地函数,主要用于结合外部接口,扩充大模型能力,解决大模型无法实时查询数据的缺点。

# 定义 GPT 调用函数,按照指定 function 来回调函数

#fun\_content 传入拟调用的函数名及参数

```
def get_gpt_call_local_func(messages, fun_content, model = "gpt-3.5-turbo-1106"):
```

```
    response = client.chat.completions.create(
        model = model,
        messages = messages,
        temperature = 0,
        seed = 1024,
```

```
        tool_choice = "auto",
        tools = [
            {
                "type": "function",
                "function": {
                    "name": fun_content.name,
                    "description": fun_content.description,
                    "parameters": fun_content.parameters
                }
            }
        ],
    )
    result = response.choices[0].message
    if (result.content is None):
        result.content = ""
    # 如果返回的是函数,则调用结果
    while (result.tool_calls is not None):
        for tool_call in result.tool_calls:
            args = json.loads(tool_call.function.arguments)
            function_to_call = globals()[tool_call.function.name]
            function_result = function_to_call(*args)
        return function_result
```

#### 3.4.2 Restful API 的设计与实现

使用 Flask 的路由特性,将 RestfulAPI 与函数进行绑定、映射,在函数中实现相关业务逻辑,实现松耦合、高内聚设计。以下是一个参考示例:

@ app.route('/upload', methods = ['POST'])

```
def upload_and_analyze():
    document_content = get_upload_doc_content(request.files
['file'])
    #准备提示词,OpenAI 按照提示词来解析文档
    .....
    analysis_result == get_completion(system_content, document_
content, output_format)
    response_data = {
        'document_content': document_content,
        'analysis_result': analysis_result
    }
    return response_data, 201
```

#### 3.4.3 外面函数接口封装

为了方便 OpenAI 与外部接口对接,将外部接口重新封装后,提供给 OpenAI 回调。以下是一个参考示例:  
# 使用“天眼查”查询公司信息

```
def search_company(keyword, token = DEFAULT_TOKEN):
    url = "http://open.api.tianyancha.com/services/open/cb/ic/2.0"
    params = {"keyword": keyword}
    headers = {"Authorization": token}
    try:
```

```
response = requests.get(url, params = params, headers = headers)
```

## 3.5 主要模块的设计与实现

### 3.5.1 项目初设模块实现

投资经理点击上传初步访谈纪要,触发 Flask 函数进行文档读取和解析。

获取初步访谈文档数据:使用 Python-docx 库解析访谈文档,提取关键信息。

构建 GPT 输入:使用访谈文章构建 GPT 输入的 prompt,根据初设项目的字段要求,形成 GPT 返回数据的规范。

调用 GPT 模型:通过 get\_gpt\_call 函数调用 OpenAI GPT 模型,将构建好的输入传递给 GPT,获取对该信息的 JSON 结果集。

页面呈现 GPT 输出:Vue 将 JSON 数据在页面上呈现。

### 3.5.2 项目公开市场尽调模块实现

项目经理完成项目立项后,系统定时进行自动的基础数据完善。

获取数据:从数据库中获取项目信息,确保获取到最新的数据。

构建 GPT 输入:根据已有的项目信息,组织 prompt,让 GPT 确定需要调用哪些函数来查询相关数据,以完成项目基础信息的自动完善。

调用 GPT 模型:利用 OpenAI GPT 模型,通过 get\_gpt\_call\_local\_func 函数调用,获取 GPT 对该信息的返回结果,其中包括确定的查询函数。

再次构建 GPT 输入:组织新的 prompt,引导 GPT 生成相应的 SQL 查询语句,以便更新数据库。

调用 GPT 模型更新数据库:再次利用 OpenAI GPT 模型,通过 get\_gpt\_call\_local\_func 函数调用,执行生成的 SQL 语句,实现项目数据库的更新。

### 3.5.3 项目初筛模块实现

投资经理在上传初筛报告后,系统触发相应的 Flask 函数进行文档读取和解析。

获取初筛报告文档数据:使用 Python-docx 库解析初筛报告文档,提取关键信息。

构建 GPT 输入:根据初筛报告内容,构建 GPT 输入的 prompt,包含项目的主要信息、评估维度等。

调用 GPT 模型:通过 get\_gpt\_call 函数调用 OpenAI GPT 模型,将构建好的输入传递给 GPT,获取对该信息的 JSON 结果集。

页面呈现 GPT 输出:调整 Vue 界面,将 GPT 生成的 JSON 数据在页面上呈现。

### 3.5.4 项目风控模块实现

风控经理根据风险评估需求,调用 Flask 函数触发相应的风险评估任务。

获取项目数据:从数据库中获取项目的相关数据,包括投前项目信息、初筛报告、公开市场尽调结果等。

构建 GPT 输入:根据风险报告数据模型的具体要求,构建 GPT 输入的 prompt,包含需要评估的各项风险因子。

调用 GPT 模型:通过 get\_gpt\_call 函数调用 OpenAI GPT 模型,将构建好的输入传递给 GPT,获取对该信息的 JSON 结果集。

页面呈现 GPT 输出:将 GPT 生成的 JSON 数据在页面上呈现。

### 3.5.5 项目投决前综合评分模块实现

综合评分模块的设计实现,主要是考虑到定时触发和手动触发(页面直接发 request),模块主要包括以下步骤:

获取管理人数据报告:从数据库中获取已完成初筛报告,且尚未进行综合评分的管理人或其他相关信息。

构建 GPT 输入:按照 GPT 评分数据表的具体要求,构建适当的 GPT 输入要求。

调用 GPT 模型:利用 OpenAI GPT 模型,通过 get\_gpt\_call 函数调用,将构建好的输入传递给 GPT 模型,以获取对该信息的 JSON 结果集。

页面呈现 GPT 输出:调整 vue 界面,将 json 集合在页面呈现。

## 4 结 语

本文成功实现了基于 GPT 的智能母基金决策投资支持系统,为投前管理、初筛、评分和风险评估等方面带来了显著的效率提升。系统上线后,帮助母基金团队深入了解被投基金的管理和投资能力,缩短了尽调时间,提高了投资效率,得到了一致好评。

然而,系统也面临一些挑战,尤其是 OpenAI 系统的不稳定、社会安全风险<sup>[9]</sup>、外部环境的不确定性。我们期望未来能基于国产大模型进行系统搭建,助力中国母基金市场发展,推动大模型在母基金领域的应用。

## 参 考 文 献

- [1] 刘震,付亮. 私募股权母基金投资要素评估研究[J]. 投资与创业,2023,34(5):7-9.
- [2] 赵宇猛. 基于 Transformer 框架的多轮对话系统模型研究[D]. 阜新:辽宁工程技术大学,2021.

- [ 3 ] 张雨乐,庄夏,戴敏. 基于 Transformer 架构的 GPT 系列模型训练技术分析[J]. 中国民航飞行学院学报,2023,34(5):16-18,22.
- [ 4 ] 杨雅清,张文帝. OpenAI 的发展路径与启示[J]. 中国电信业,2023(7):52-55.
- [ 5 ] 孙小鱼. 一种对话机器人开发技术综述与系统架构实现[J]. 软件工程,2021,24(2):45-48.
- [ 6 ] Smyth P. Creating web APIs with Python and flask[J]. The Programming Historian,2018,72(4):1121-1129.
- [ 7 ] 胡晓萌,陈楚仪. AI Agent, 为什么是 AIGC 最后的杀手锏? [EB/OL]. [2023-11-20]. [https://mp.weixin.qq.com/s/?\\_\\_biz=MjM5OTE0ODAxMjQ=&mid=2650973217&idx=1&sn=8d0d9b2f159635d387664f02d6cc7f4d&chksm=bc9eb538bbe6245cb2c1243d9af169ad6907bc17c0121a8c0ccd25f19dc442386c7160bb877#rd](https://mp.weixin.qq.com/s/?__biz=MjM5OTE0ODAxMjQ=&mid=2650973217&idx=1&sn=8d0d9b2f159635d387664f02d6cc7f4d&chksm=bc9eb538bbe6245cb2c1243d9af169ad6907bc17c0121a8c0ccd25f19dc442386c7160bb877#rd).
- [ 8 ] OpenAI. Function calling[EB/OL]. [2023-11-20]. <https://platform.openai.com/docs/guides/function-calling>.
- [ 9 ] 简圣宇. GPT 语言模型的心智问题、影响与风险——从 ChatGPT 谈起[J]. 上海大学学报(社会科学版),2023,40(6):63-78.
- [ 1 ] 展趋势[J]. 节能技术,2006,34(6):534-537.
- [ 2 ] 许相波. 大型火电机组热经济性的在线计算[D]. 太原:太原理工大学,2008.
- [ 3 ] Rosen M A, Dincer I. Exergoeconomic analysis of power plants operating on various fuels[J]. Applied Thermal Engineering, 2003, 23(6):643-658.
- [ 4 ] 冯少山. 300MW 机组汽轮机及其辅助系统节能诊断[D]. 广州:华南理工大学,2017.
- [ 5 ] 石勇,崔焕顺,李涛永. 大型火电厂经济指标考核管理系统[J]. 电力科学与工程,2008,24(10):53-56.
- [ 6 ] 叶永松,王莉琳,吴耀凯,等. 以能耗和安全为基础的火电厂小指标考核研究[J]. 华中电力,2007(5):16-18.
- [ 7 ] 郭民臣,魏楠. 火电机组性能监测与小指标在线考核管理系统在电厂运行优化方面的应用[J]. 电力设备,2007(2):12-15.
- [ 8 ] 王莉琳,叶永松,汪家军. 火电机组运行小指标考核方法探讨[J]. 热力发电,2004(6):39-41.
- [ 9 ] 徐广震. 基于耗差分析理论的火电机组运行优化指导系统研究[D]. 北京:华北电力大学,2008.
- [ 10 ] 张春发,张宝,孙伟,等. 基于 C/S 模式的火电机组耗差分析系统[J]. 汽轮机技术,2002,12(4):193-195.
- [ 11 ] 王松浩. 耗差分析方法的现状及其发展研究[J]. 江苏科技信息,2017(32):25-27.
- [ 12 ] 陈龙. 等效焓降法与常规热平衡法在热力系统的计算分析[J]. 能源研究与利用,2015(4):26-30.
- [ 13 ] 赵晓峰. 电厂热经济分析中重要参数耗差计算模型的建立[D]. 北京:华北电力大学,2009.
- [ 14 ] 郑楠,叶学民. 基于等效焓降法的烟气余热利用热经济性分析[J]. 电力科学与工程,2020,36(7):74-78.
- [ 15 ] 杨静. 循环函数法在供热机组性能计算中的应用[J]. 区域供热,2013(6):32-35.
- [ 16 ] 马芳礼. 电厂热力系统节能分析原理:电厂蒸汽循环的函数与方程[M]. 北京:水利电力出版社,1992.
- [ 17 ] 郭民臣,王清照,魏楠,等. 电厂热力系统矩阵分析法的改进[J]. 热能动力工程,1997,11(2):24-27.
- [ 18 ] 王加璇,王清照,宋乃辉. 热经济学研究的使命与任务[J]. 热能动力工程,2002,21(2):111-114.
- [ 19 ] Valero A, Lozano M A. On-line monitoring of power plant performance using exergetic cost techniques [J]. Applied thermal engineering,1996,16(12):933-948.
- [ 20 ] 胡文杰. 火电机组耗差分析及软件系统开发[D]. 北京:华北电力大学,2015.
- [ 21 ] 钱瑾,王培红,李琳. 聚类算法在锅炉运行参数基准值分析中的应用[J]. 中国电机工程学报,2007(23):71-74.
- [ 22 ] 王会强. 660MW 超超临界火电机组耗差分析系统的开发与应用[D]. 北京:华北电力大学,2016.
- [ 23 ] 白尊亮. 基于循环函数法的火电机组耗差分析模型及软件的研究[D]. 上海:上海交通大学,2010.

## (上接第 8 页)

行参数特点及运行方法的不同,给出三类目标值的获取方法。第一类采用机组设计值;第二类采用 K-means 聚类算法,通过对历史数据进行聚类计算得出不同工况下的聚类值,再将聚类值进行拟合即可得到回归方程;第三类为变工况计算,根据机组实际运行工况计算得到相应的最佳运行值。通过以上三种方案可为机组运行参数获取准确有效的目标值。

2) 锅炉侧建立锅炉效率偏差模型,汽机侧建立热力系统循环函数模型,通过计算不同加热单元的循环排气系数并结合矩阵表达式,基于某电厂 600 MW 现场实际数据进行耗差计算得出热经济参数,并分析得出影响煤耗增大的具体原因。通过做出主要影响参数的典型工况运行优化曲线,从而可直观看出参数的实际运行值与目标值,以及参数偏离目标值造成的煤耗变化量,避免了小指标考核体系孤立的问题,为值班员进行参数调整提供参考指导。

3) 基于 B/S 架构,采用 JAVA 语言开发了具有参数设置、数据查询、耗差计算结果查询等功能的信息平台,可对机组实际运行过程中的各项参数进行不间断监视和分析,从而有效指导运行人员针对性地采取措施保证机组能在最优状态运行,为电厂节能降耗以及经济运行提供了平台依托。

## 参 考 文 献

- [ 1 ] 于淑梅,张文宝. 电厂热力系统分析方法的研究现状及发