

电流迁移态序列特征模型与获取方法研究*

张慧^{1,2}, 刘帅^{1,2}, 杨泽丞^{1,2}, 王萍^{1,2}, 程红梅^{1,3}, 张振亚^{1,2†}

(1. 智能建筑与建筑节能安徽省重点实验室 安徽建筑大学, 安徽 合肥 230022; 2. 安徽建筑大学 电子与信息工程学院, 安徽 合肥 230022; 3. 安徽建筑大学 经济与管理学院, 安徽 合肥 230022)

摘要: 基于使用智能插座对电路中的电流高频次监测、获得用户用电态势数据, 对电流迁移态序列、电流迁移态序列特征的一元回归模型进行了研究. 基于电流稳态序列 ε 片段、电流稳态序列片段, 设计了电流迁移态序列的构造方法, 提出使用一元回归模型描述每一个电流迁移态序列, 实现将长度不定的电流迁移态序列到维数固定的电流迁移态序列特征空间的映射. 进一步, 设计了基于微环境的粒子群优化算法(Microenvironment based Particle Swarm Optimization, MPSO), 实现了电流迁移态序列一元回归特征的优化. 实验表明: 使用所提电流迁移态序列特征进行电器状态识别, 平均可以达到97.93%的准确率, 且相较PSO算法与CAPSO算法, MPSO算法在使用较少的粒子数达到与这两种算法一致精度的同时, 使用时间显著降低.

关键词: 电流迁移态序列; 粒子群优化算法; 一元回归模型; 微环境

DOI: 10.13568/j.cnki.651094.651316.2023.06.29.0001

中图分类号: TP391.4 **文献标识码:** A **文章编号:** 2096-7675(2024)01-0037-015

引文格式: 张慧, 刘帅, 杨泽丞, 王萍, 程红梅, 张振亚. 电流迁移态序列特征模型与获取方法研究[J]. 新疆大学学报(自然科学版)(中英文), 2024, 41(1): 37-51.

英文引文格式: ZHANG Hui, LIU Shuai, YANG Zecheng, WANG Ping, CHENG Hongmei, ZHANG Zhenya. Research on feature model and mining method for current transition sequence[J]. Journal of Xinjiang University(Natural Science Edition in Chinese and English), 2024, 41(1): 37-51.

Research on Feature Model and Mining Method for Current Transition Sequence

ZHANG Hui^{1,2}, LIU Shuai^{1,2}, YANG Zecheng^{1,2}, WANG Ping^{1,2},
CHENG Hongmei^{1,3}, ZHANG Zhenya^{1,2}

(1. Anhui Province Key Laboratory of Intelligent Building & Building Energy Saving, Anhui Jianzhu University, Hefei Anhui 230022, China; 2. School of Electronic and Information Engineering, Anhui Jianzhu University, Hefei Anhui 230022, China; 3. School of Economics and Management, Anhui Jianzhu University, Hefei Anhui 230022, China)

Abstract: In this paper, based on the high-frequency monitoring of current in the circuit via smart sockets and electrical situation data of the customer obtained, the current transition sequence and the univariate regression model for the feature of the current transition sequence are proposed. Based on the ε -fragment for the current steady sequence and fragment for the current stable sequence, a mining method for current transition sequence discovery is designed. And further, a univariate regression model is proposed to describe each current transition sequence. And with the univariate regression model, each current transition sequence in variable length is mapped into the feature space for the current transition sequence with fixed dimension. Furthermore, a microenvironment-based particle swarm optimization (MPSO) algorithm is given to optimize the univariate regression features for the current transition sequences. The experimental results show that using the current transition sequence features proposed in this paper for electrical device state recognition can achieve an average accuracy of 97.93%. Compared with the PSO algorithm and CAPSO algorithm, the MPSO algorithm used in this paper achieves the same accuracy with fewer particles and significantly reduces usage time.

* 收稿日期: 2023-06-29

基金项目: 安徽省高校自然科学研究重点项目“面向行为节能的室内人员敏感行为获取模型研究”(KJ2020A0470); 安徽省特支计划创新领军人才项目(皖组办[2022]21号).

作者简介: 张慧(1999—), 女, 硕士生, 从事建筑模式识别与智能信息处理的研究, E-mail: 1819218868@qq.com.

† 通讯作者: 张振亚(1972—), 男, 博士, 教授, 主要从事计算智能、数据挖掘的研究, E-mail: zhenyazhang@ustc.edu.cn.

Key words: current transition sequence; particle swarm optimization algorithm; univariate regression model; microenvironment

0 引言

电是现代生产、生活不可替代的能源形式,用电过程的管控事关用电安全,一直为行业应用所关注.随着物联网技术的普及,精细的用电监测与管理日益为越来越多的研究者所关注^[1-3].目前,对于用户需求侧的用电研究,主要是基于电力负载特征,如电压、电流、功率等,对电器进行稳态分析、暂态分析以及两者混合分析的方式进行^[4].文献[5]从电器电流波形中提取了电流峰值、电流均值、电流均方值、高电平比、低电平比等10个时域和稳态特征,采用k-最近邻规则(k-NNR)和支持向量机进行负载识别,并将主成分分析与k-NNR和SVM相结合分别对10个特征进行检验,选择出最有效的特征.文献[6]使用智能插座采集电器用电数据,从负载功率序列样本中提取负载的功率特征和人的行为特征,提出了一种基于贝叶斯分类模型的家用电器识别算法.文献[7]基于V-I轨迹的信息,如面积、斜率、曲率、轨迹质心等构建稳态和暂态特征.文献[8]从用电数据中构建了时间和频谱负载特征,并提出了双流卷积神经网络(TSCNN)从两种类型的负载特征中提取特征并执行分类.相比稳态特征,暂态特征与设备本身的特性相关性更强^[9-10],且用电设备之间的暂态特征重叠较少,有利于用电负荷的分解.对于暂态特征的提取,常使用信号分析方法将高频采集的用电数据转换为频域或小波域.短时傅里叶变换(STFT)和快速傅里叶变换(FFT)广泛用于提取电流谐波,有助于区分暂态负载特征^[11-12].文献[13]和文献[14]基于小波变换进行特征的提取,可以在频域和时域上表示信号,从而允许访问有关信号的局部信息.文献[15]提出使用S变换来提取设备特征,然后将这些特征输入SVM分类器以识别每个电气设备,也取得了较高的分类准确度.但高采样频率对硬件设备以及计算能力均提出了更高的要求,增加了设备成本和计算成本.电流作为一种常见的用户用电态势数据,其实时变化能够及时地反映用电设备状态的变化,通过对电流迁移状态的研究,有助于了解设备运行状态的切换,为用户提供电力使用和管理方面的指导和决策支持.为了快速、有效地提取用电事件特征,本文提出了一种电流迁移态序列特征模型与获取方法,在使用与设备本身特性相关性更强的迁移态特征的基础上,相比信号分析方法计算量较小,能够以较低的成本满足快速、有效地提取电流迁移态序列特征的需求,保证电器工作模式在线辨识的及时性.

由于电器通过智能插座取电时,不仅可以用户使用智能插座实现对电器的更多操作,也使得电器用电数据的获取更为及时、有效、简洁.本文在使用自主研发的智能插座高频次采集电器^[16-17]用电数据的基础上,提出了一种基于微环境的分布式粒子群优化算法(MPSO)的电流迁移态特征提取方法.该方法在构造电流迁移态序列集的基础上,使用一元回归模型描述电流迁移态序列的特征,并使用MPSO算法对电流迁移态序列的特征进行了求解,快速、有效地实现了对电流迁移态序列特征的提取.本文主要贡献:1)根据电流迁移态的数据变化特性提出了一种电流迁移态序列的定位方法;2)设计了电流迁移态序列的一元回归模型特征;3)提出基于微环境的MPSO算法,优化电流迁移态序列特征.

1 电流迁移态特征提取方法

电器在使用时,其工作状态经常从一种状态变化到另外一种状态.通常,当电器稳定在某一工作状态时,电路上的用电态势数据相对稳定;而当电器的工作状态发生变化时,电路上的用电态势数据相应变化.简单的,称电器稳定工作的状态为稳态,相应的,对处于稳态的电器用电情况进行在线监测所获得的电器用电情况的状态序列称为稳态序列.对时间上邻近的两个稳态,电器的工作状态从前一个稳态迁移到后一个稳态时,一般不能够从一个稳态瞬时进入下一个稳态,而是需要一定的时间间隔,称这段时间间隔内对电器用电情况进行在线监测所获得的电器用电情况的状态序列为迁移态序列,相应的,称这段时间内电器所处的状态为迁移态.电器的工作/使用过程,是其在稳态与迁移态之间不断切换的过程.

电流值是一类基础的电态势数据,基于高频次采集的电流数据迁移态特征提取是用户侧用电研究中经常面

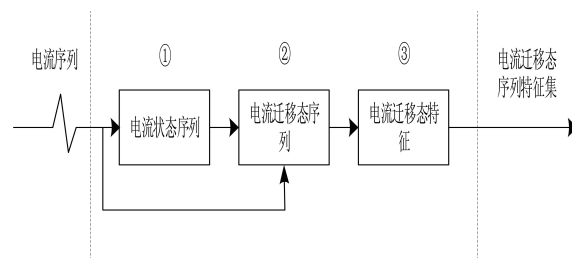


图 1 电流迁移态特征提取

对的问题. 电流迁移态特征提取流程如图1所示.

图1示意的流程主要包含三个部分: 电流状态序列模块①、电流迁移态序列模块②、电流迁移态特征模块③. 电流状态序列模块①对电流序列进行深度解析, 析构出电流序列中的每一个稳态序列; 电流迁移态序列模块②以电流状态序列模块①输出的稳态序列以及电流序列为输入, 析构出蕴含于电流序列中的电流迁移态序列; 电流迁移态特征模块③负责将电流迁移态序列模块②输出的电流迁移态序列转换为电流迁移态特征.

定义 1 用电流序列与电流状态序列 ε 片段

设 $t_0, t_n \in R, t_0 > 0, t_n > 0, t_0 < t_n, T = (t_0, t_n]$ 为观测时间段, 对 $t_0 < t_1 < \dots < t_i < \dots < t_n$, 需要在 $(t_{i-1}, t_i]$ 时间段内观测一次电路中的电流. 又设 s_i 是 $(t_{i-1}, t_i]$ 时间段内观测到的电流值, $i = 1, 2, \dots, n$, 称 $S = \langle s_1, s_2, \dots, s_n \rangle$ 为在 $T = (t_0, t_n]$ 时间段内的用电电流序列. 进一步, 设 $es_l^{(\varepsilon)} = \langle s_l, s_{l+1}, \dots, s_{l+m} \rangle$, $es_l^{(\varepsilon)} \subseteq S$, 是 $\Delta_l^m = (t_l, t_{l+m}]$ 时间段内的用电电流序列, m 是 Δ_l^m 时间段内观测到的电流值个数, $\bar{s} = [1/(m+1)] \sum_{i=l}^{l+m} s_i$, $\varepsilon > 0$. 若不等式组 (1) 成立, 称 $es_l^{(\varepsilon)}$ 是一个电流状态序列 ε 片段, 电器在 Δ_l^m 时间段内电流状态处于 ε 稳态, $es_l^{(\varepsilon)}$ 是一个电流状态序列 ε 片段.

$$\begin{cases} |\bar{s} - s_i| \leq \varepsilon \\ l \leq i \leq l+m \\ |\bar{s} - s_{l+m+1}| > \varepsilon \\ |\bar{s} - s_{l-1}| > \varepsilon \end{cases} \quad (1)$$

不等式组 (1) 设定的是一种基于均值的电流状态序列 ε 片段. 实践中可以由类似不等式组 (2) 的其它约束条件规定电流状态序列 ε 片段, 这里, 不等式组 (2) 设定的是一种基于跃变检测的序列 ε 片段: 在一个电流状态序列 ε 片段中, 任意两个相邻观测到的数据, 其差异值不大于 ε .

$$\begin{cases} |s_i - s_{i+1}| \leq \varepsilon \\ l \leq i \leq l+m-1 \end{cases} \quad (2)$$

设 $es_{i_1}^{(\varepsilon)}, es_{i_2}^{(\varepsilon)}, \dots, es_{i_u}^{(\varepsilon)}$ 是 S 中全部电流状态序列 ε 片段, $i_1 < i_2 < \dots < i_u$. 又设 $es_{i_j}^{(\varepsilon)}$ 对应的时间段为 τs_{i_j} , $1 \leq j \leq u$, 若 $\tau s_{i_j} \cap \tau s_{i_{j+1}} = \Phi$, $1 \leq j \leq u-1$, 称 $ES^{(\varepsilon)} = \{es_{i_1}^{(\varepsilon)}, es_{i_2}^{(\varepsilon)}, \dots, es_{i_u}^{(\varepsilon)}\}$ 是 S 中全部电流状态序列 ε 片段构成的电流状态序列 ε 片段集.

定义 2 电流状态序列片段

对 S 中全部电流状态序列 ε 片段构成的电流状态序列 ε 片段集 $ES^{(\varepsilon)} = \{es_{i_1}^{(\varepsilon)}, es_{i_2}^{(\varepsilon)}, \dots, es_{i_u}^{(\varepsilon)}\}$, $i_1 < i_2 < \dots < i_u$, $es_{i_j}^{(\varepsilon)}$ 对应的时间片段为 τs_{i_j} . 令 $\tau s_{i_j} = (\tau s_{i_j, \text{begin}}, \tau s_{i_j, \text{end}}]$, 对 τs_{i_i} 与 $\tau s_{i_{i+1}}$, 若 $\tau s_{i_i, \text{end}} = \tau s_{i_{i+1}, \text{begin}}$ 且 $|\bar{es}_{i_i} - \bar{es}_{i_{i+1}}| \leq \Delta > 0$, 需要将 $es_{i_i}^{(\varepsilon)}$ 与 $es_{i_{i+1}}^{(\varepsilon)}$ 合并为一个序列片段, 称为电流状态序列片段; 若不满足, 则称 $es_{i_i}^{(\varepsilon)}$ 与 $es_{i_{i+1}}^{(\varepsilon)}$ 分别为一个电流状态序列片段.

对 T 时间段内的观测电流序列 $S = \langle s_1, s_2, \dots, s_n \rangle$, $\varepsilon > 0, \Delta > 0$, 设 S 中蕴含了 k 个电流状态序列片段, es_i 是第 i 个电流状态序列片段, $i = 1, \dots, k$, 称 $ES = \{es_1, es_2, \dots, es_k\}$ 是 S 中全部电流状态序列片段构成的电流状态序列片段集.

易证, 任意两个电流状态序列片段, 其对应发生时间段的交集为空.

定义 3 电流迁移态序列

设 $S = \langle s_1, s_2, \dots, s_n \rangle$ 为 $T = (0, t]$ 时间段内的观测电流序列, $ES = \{es_1, es_2, \dots, es_k\}$ 为 S 中全部电流状态序列片段构成的电流状态序列片段集, es_i ($1 \leq i \leq k$)是电流状态序列片段, es_i 对应的时间片段为 $\tau s_i = (\tau s_{i, \text{begin}}, \tau s_{i, \text{end}}]$, $\tau s_{i, \text{begin}} \geq 0, \tau s_{i, \text{end}} \geq 0, \tau s_{i, \text{begin}} \leq \tau s_{i, \text{end}}$. 称 es_i 为一个电流稳态序列, $1 \leq i \leq k$, 相应的, 电器在 $(\tau s_{i, \text{begin}}, \tau s_{i, \text{end}}]$ 时间段内的运行状态为电流稳态, 简称稳态. 进一步, 令 $\tau s_{0, \text{end}} = 0, \tau s_{k+1, \text{begin}} = t, move_i$ 为 $(\tau s_{i-1, \text{end}}, \tau s_{i, \text{begin}}]$ 对应的电流序列, 称 $move_i$ 为一个电流迁移态序列, 相应的, 电器在 $(\tau s_{i-1, \text{end}}, \tau s_{i, \text{begin}}]$ 时间段内的工作状态为电流迁移态, 简称迁移态.

电流迁移态序列的获取是电流迁移态特征提取的基础, 图2给出了电流迁移态序列的生成框架, 该框架是图1中的模块①和模块②的细化. 图1示意的电流迁移态特征提取流程中, 模块①和模块②用于电流迁移态序列的确定.

图2示意的电流迁移态序列生成框架中:首先,基于类似不等式组(1)的约束条件的电流稳态判别策略 ε 被用来从样本电流序列中解构出全部的电流状态序列 ε 片段;接着,全部的电流状态序列 ε 片段被片段合并策略 Δ 处理生成全部的电流状态序列片段集作为样本电流序列中全部的电流稳态序列;迁移态生成策略在接收到电流状态序列片段集后,将每个电流状态序列片段从样本电流序列中消除,获得全部的电流迁移态序列.关于一个刚刚获得的电流观测值是否属于当前电流状态序列 ε 片段的判别策略在算法1中给出.算法1依据不等式组(2)设计:若待判别的电流观测值与当前电流状态序列 ε 片段最后一个数据差异较大,待辨识的电流观测值不属于当前电流状态序列 ε 片段.

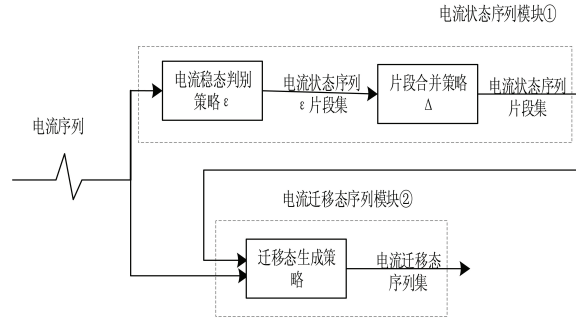


图 2 电流迁移态序列生成框架

算法 1 基于跃变检测的电流状态序列 ε 片段辨识策略stableJudge

输入: s , 待检测数据; es , 当前电流状态序列片段; 阈值 ε

输出: $Flag$, 待检测数据是否隶属于电流序列片段

- 1) $data = \text{tail}(es)$; //将 es 的最后一个数据赋值到 $data$ 中
- 2) $M = \text{abs}(s - data)$; //计算 $data$ 与 s 差的绝对值
- 3) if $M > \varepsilon$
 $Flag = 0$; //不属于当前序列 ε 片段
- 4) else
 $Flag = 1$; //属于当前序列 ε 片段

以算法1为辨识策略,电流序列 S 中的全部电流状态序列 ε 片段获取的流程由算法2给出.算法2通过从头到尾逐一对每个电流数据使用辨识策略stableJudge获取 S 中的每一个电流状态序列 ε 片段,并保存到全部电流状态序列 ε 片段集 εES 中.

算法 2 电流状态序列 ε 片段集生成算法

输入: S , 电流序列; 阈值 ε

输出: εES , 蕴含于 S 中全部电流状态序列 ε 片段的集合

- 1) $\varepsilon ES = \phi$; // εES 置空
- 2) $n = \text{length}(S)$; // S 中电流数据的数量
- 3) $es = \phi$
- 4) for $i = 1 : 1 : n$
- 5) $s = S(i)$
- 6) if $es == \phi$
- 7) $es = es \cup \{s\}$
- 8) else
- 9) $data = \text{tail}(es)$; //将 es 的最后一个数据赋值到 $data$ 中
- 10) $flag = \text{stableJudge}(es, data, \varepsilon)$; //调用算法1判断 $data$ 是否应该加入到 es 中
- 11) if $flag == 0$
- 12) if $\text{length}(es) > 1$
- 13) $\varepsilon ES = \varepsilon ES \cup \{es\}$

- 14) $es == \phi$
- 15) else
- 16) $es = es \cup \{s\}$
- 17) if $es \neq \phi$ and $\text{length}(es) > 1$
- 18) $\varepsilon ES = \varepsilon ES \cup \{es\}$

图2中,电流状态序列 ε 片段集由片段合并策略 Δ 生成,算法3描述了合并策略 Δ 的流程. `getFirstElement()`用于获取电流状态序列 ε 片段集的首个元素,即第一个电流状态序列 ε 片段; `isAdjacent()`用于判别两个 ε 片段集在电流序列 S 中是否相邻,相邻时返回值为1,否则为0,且最后一个参数为空时,返回值亦为0; `mean()`返回电流状态序列 ε 片段的平均值; `concat()`用于将候选电流状态序列片段的前部和尾部拼接成新的候选电流状态序列片段的前部.

算法 3 电流状态序列 ε 片段集生成算法(片段合并策略 Δ)

输入: S , 电流序列; εES , 蕴含于 S 中的全部电流状态序列 ε 片段的集合; Δ , ε 片段合并阈值

输出: ES , 蕴含于 S 中全部电流状态序列片段的集合

- 1) $ES = \phi$; // ES 置空
- 2) $esHead = \text{getFirstElement}(\varepsilon ES)$; // 获取 εES 序列的第一个元素,该元素是一个电流状态序列 ε 片段
// $esHead$ 是候选电流状态序列片段的前部
- 3) $esTail = \phi$; // $esHead$ 是候选电流状态序列片段的尾部,初始值为空
- 4) $\varepsilon ES = \varepsilon ES - \{esHead\}$; // 从电流状态序列 ε 片段的集移除第一个元素
- 5) while $\varepsilon ES \neq \phi$ // 以下6)~17)步需要重复执行,直至 εES 为空
- 6) $esTail = \text{getFirstElement}(\varepsilon ES)$; // εES 非空时获取序列的第一个元素
- 7) $\varepsilon ES = \varepsilon ES - \{esTail\}$; // 从电流状态序列 ε 片段的集移除第一个元素
- 8) $flag = \text{isAdjacent}(S, esHead, esTail)$; // 判断 $esHead$ 、 $esTail$ 在 S 中是否相邻
// $flag=0$ 时不相邻, $flag=1$ 时相邻
- 9) if $flag == 0$ or $(flag == 1 \text{ and } \text{abs}(\text{meanHead} - \text{meanTail}) > \Delta)$
// $esHead$ 、 $esTail$ 不相邻或者均值差异大于 Δ
// ε 片段 $esHead$ 是一个完整的电流状态序列片段,将其加入全部电流状态序列片段的集合 ES
- 10) $ES = ES \cup \{esHead\}$; // ES 新增一个元素
- 11) $esHead = esTail$; // 更新候选电流状态序列片段的前部
- 12) $esTail = \phi$; // 若 εES 非空,候选电流状态序列片段的尾部在步骤6)更新
// 这里置空,是为了便于循环退出后的扫尾处理
- 13) else // for $flag == 1$ and $\text{abs}(\text{meanHead} - \text{meanTail}) \leq \Delta$
// $meanHead$ 、 $meanTail$ 属于同一电流状态序列片段,需要合并成其所在电流状态序列片段的前部
- 14) $meanHead = \text{mean}(esHead)$
- 15) $meanTail = \text{mean}(esTail)$
- 16) $esHead = \text{concat}(esHead, esTail)$
- 17) $esTail = \phi$; // 若 εES 非空,候选电流状态序列片段的尾部在步骤6)更新
// 这里置空,是为了便于循环退出后的扫尾处理
- 18) $ES = ES \cup \{esHead\}$

在使用算法3获得全部电流状态序列片段集后,依据定义3,将每个代表电器处于稳定工作状态时的电流状态序列片段从电流序列片段中剔除,即获得电流序列中全部电流迁移态序列片段,实现电流迁移态序列片段集的获取. 电流迁移态序列片段集获取算法(生成策略)在算法4中给出.

算法 4 电流迁移态序列片段集获取算法(生成策略)

输入: S , 电流序列; ES , 蕴含于 S 中全部电流状态序列片段的集合

输出: TS , 蕴含于 S 中全部电流迁移态序列片段的集合

- 1) $TS = \phi$; // TS 初始化为空
- 2) $tsBegin = 1$; // ts 用于暂存一个候选电流迁移态序列片段, $tsBegin$ 是 ts 在电流序列 S 中的位置
// 电流序列 S 第一个数据的位置为1
- 3) while $ES \neq \phi$
- 4) $es = \text{getFirstElement}(ES)$; // es 是 ES 中的第一个电流状态序列片段
// 用于缓存候选电流迁移态序列片段生成时参考电流状态序列片段
- 5) $ES = ES - \{es\}$; // ES 保存了全部未使用的电流状态序列片段
- 6) $locBegin = \text{Location}(es, S, begin)$; // $locBegin$ 指示 es 首个数据在 S 中的位置
- 7) $tsEnd = locBegin - 1$; // ts 最后一个数据在 S 中的位置
- 8) $ts = S(tsBegin:tsEnd)$; // 从 S 中析构出候选电流迁移态序列片段
- 9) $TS = TS \cup \{ts\}$; // 将候选电流迁移态序列片段加入到电流迁移态序列片段集
- 10) $tsBegin = \text{Location}(es, S, end) + 1$; // $\text{Location}(es, S, end)$ 是 es 最后一个数据在 S 中的位置
// 更新下一个候选电流迁移态序列片段第一个数据在 S 中的位置
- 11) $ts = S(tsBegin:\text{length}(S))$
- 12) $TS = TS \cup \{ts\}$

为高效获得电流迁移态序列片段集, 无论是电流状态序列 ϵ 片段集还是电流状态序列片段集, 其中作为集合元素的每一个片段都按照其首个数据在电流序列集中的位置从前到后渐次存储. 需要注意的是, 由于用电设备的不稳定性, 可能使得迁移态序列较为复杂, 实际应用过程中, 需要进一步排除不稳定状态对电器工作状态辨识的影响.

电器运行时从一个稳定状态迁移到另外一个稳定状态虽然一般不需要太多的时间, 但也存在一定的波动, 这意味着一个迁移态序列中数据的个数是变化的. 为统一描述每一个迁移态, 将每个迁移态序列进行了一元线性回归处理, 一元线性回归方程的两个常系数作为该迁移态序列的特征表示.

设 $S = \langle s_1, s_2, \dots, s_n \rangle$ 是一个电流迁移态序列片段, $S' = \langle s'_1, s'_2, \dots, s'_n \rangle$ 是一元线性回归模型对序列 S 中每个值的估计, 成立:

$$\begin{cases} s'_k = s'_{k-1} + a, & k > 1, a \in \mathbb{R} \\ s'_1 = b, & k = 1, b \in \mathbb{R} \end{cases} \quad (3)$$

对每个电流迁移态序列片段, 式(3)常系数 (a, b) 的最优值即是该序列的特征. 规定 $E(S, S')$ 如式(4), 则对电流迁移态序列片段 S , 其特征 (a, b) 是式(5)描述最优化问题的最优解, 如式(3)中的常系数 a 和 b .

$$E(S, S') \stackrel{d}{=} \frac{1}{2n} \sum_{i=1}^n (s_i - s'_i)^2, \quad s_i \text{ 不缺失} \quad (4)$$

$$E(S, S^*) = \min E(S, S') \quad (5)$$

2 基于MPSO算法的特征求解

用电数据采集过程中, 出现了约11%的电流数据缺失, 针对这种情形, 使用粒子群优化算法(Particle Swarm Optimization, PSO)求解电流迁移态序列片段 S 一元线性回归模型常系数.

PSO算法中, 设 t 时刻某粒子位置在 $x(t)$ 、速度为 $v(t)$, 则 $t+1$ 时刻, 其位置 $x(t+1)$ 与速度 $v(t+1)$ 一般按照式(6)、式(7)规定的策略更新. 式(6)中, ω 是惯性因子, c_1 、 c_2 是学习因子, $\text{rand}()$ 是随机数, $pBest$ 是粒子的历史最优位置, 而 $gBest$ 是全部粒子的历史最优位置. 相应的算法流程在算法5中给出.

$$v(t+1) = \omega \times v(t) + c_1 \times \text{rand}() \times (pBest - x(t)) + c_2 \times \text{rand}() \times (gBest - x(t)) \quad (6)$$

$$x(t+1) = x(t) + v(t+1) \quad (7)$$

算法 5 PSO算法

输入: 粒子数 n , 粒子位置维度 d , 学习因子 c_1 、 c_2 , 惯性因子 ω , 最大迭代次数 $\max N$, 适应度函数 fun

输出: 最优个体的位置 $gBest$

- 1) 依据粒子位置维度 d , 初始化每个粒子的速度与位置
- 2) 将每个粒子的初始位置作为其历史最优位置 $pBest$ 的初始值
- 3) 计算全部粒子最优位置 $gBest$
- 4) $k=1$
- 5) while $k \leq \max N$
- 6) 获取每个邻居节点的历史最优位置 $pBest$
- 7) 依据式 (3) 计算 $gBest$
- 8) for each 粒子
- 9) 依据式 (6) 和式 (7) 更新速度和位置
- 10) 使用函数 fun 计算位置更新后粒子的适应度值
- 11) 若新位置的适应度值较粒子历史最佳适应度的值更小, 将粒子历史最优位置设置为粒子的当前位置
- 12) 比较全部粒子最优位置 $gBest$ 和每个粒子的历史最优位置的适应度值, 将适应度值最小的更新全部粒子最优位置 $gBest$
- 13) $k = k + 1$
- 14) 输出 $gBest$

由于PSO算法是一种基于种群的随机优化算法, 所生成的解是最优解的逼近, 对具体的最优化问题, 一般需要使用PSO算法重复多次执行获得一组最优解的逼近后, 从中选取最优的作为问题的最优解, 这种方式将显著增加最优化问题求解的时间.

为加速使用PSO算法求解最优化问题, 对粒子群优化算法, 本文所有的粒子划分成若干个组, 一个包含若干个粒子的组被视为一个微环境: 1) 微环境内, 以微环境内的粒子为种群, 可以独立执行适应微环境信息特性、类似算法5的PSO算法; 2) 一个微环境内全部粒子的历史最优位置保存在微环境内; 3) 一个微环境与若干个微环境可以进行信息交互, 互相查询被查询微环境内的历史最优的粒子的位置信息; 4) 从信息交互的角度看, 全部微环境形成了一个以微环境为节点的微环境网络; 5) 微环境网络中, 邻接的微环境节点可以进行信息交互, 不相邻的微环境节点不能够进行信息交互. 图3示意了一个微环境网络拓扑结构.

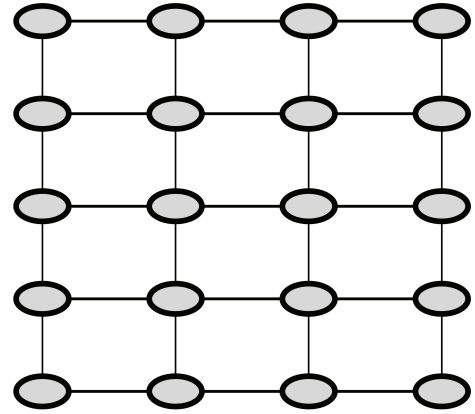


图3 微环境网络拓扑结构

图3中, 每个灰色圆点表示一个微环境节点, 边表示了微环境节点的连接(信息交互)情况. 逻辑上, 全部微环境节点以5行4列形式排列, 共20个微环境节点. 每个微环境节点只能够向直接邻接的微环境节点发送查询请求, 任意两个不直接连接的微环境节点不能直接交换数据.

相应的, 对基于微环境的PSO算法, 设: 1) 每个微环境中有若干个粒子; 2) 对任意微环境, 记 $gBest_L$ 是该微环境中全部粒子的历史最优位置; 3) 对某个微环境, 若其有 m 个邻居微环境, 记 $gBest_{L_i}$ 是其第 i 个邻居微环境中全部粒子的最优位置, 规定 $gBest_A$ 如式 (8). 则对基于微环境的PSO算法, 某个微环境中任意粒子, 若 t 时刻其位置为 $x(t)$ 、速度为 $v(t)$, 则其在 $t+1$ 时刻的速度 $v(t+1)$ 与位置 $x(t+1)$ 可以依据式 (9)、式 (7) 确定. 称式 (8) 是基于微环境的PSO算法的数据交换策略, 而式 (9)、式 (7) 是基于微环境的PSO算法中粒子位置的更新策略. 相应的算法流程在算法6中给出.

$$gBest_A = \min(gBest_L, gBest_{L_1}, gBest_{L_2}, \dots, gBest_{L_m}) \quad (8)$$

$$v(t+1) = \omega \times v(t) + c_1 \times rand() \times (pBest - x(t)) + c_2 \times rand() \times (gBest_A - x(t)) \quad (9)$$

算法 6 基于MPSO算法

输入: 本地粒子数 n , 粒子位置维度 d , 学习因子 c_1 、 c_2 , 惯性因子 ω , 最大迭代次数 $\max N$, 适应度函数 fun , 邻居微环境节点列表 $pNeighbor$

输出: 本地全部粒子的历史最优位置 $gBest_L$

- 1) 初始化每个本地粒子的速度与位置
- 2) 将每个粒子的初始位置作为其历史最优位置的初始值
- 3) 计算本地全部粒子最优位置 $gBest_L$
- 4) $k=1$
- 5) while $k \leq \max N$
- 6) 对 $pNeighbor$ 中每个邻居微环境节点查询其历史最优位置 $gBest_{Li}$, $i=1,2,\dots,m$,
 m 为邻居微环境节点个数
- 7) 依据式(8)计算 $gBest_A$
- 8) for each本地粒子
- 9) 依据式(9)和式(7)更新速度和位置
- 10) 使用函数 fun 计算位置更新后粒子的适应度值
- 11) 若新位置的适应度值较粒子历史最佳适应度值更小, 将粒子历史最优位置设置为粒子当前位置
- 12) 更新本地全部粒子最优位置 $gBest_L$
- 13) $k = k + 1$
- 14) 输出 $gBest_L$

算法6中, 以粒子的适应度值评估每个粒子位置的优劣: 适应度值越小, 粒子位置越优越; 粒子的适应度值用适应度函数 fun 计算, 该函数以粒子的位置为自变量.

设 $gBest$ 是 k 时刻微环境网络中全部微环境全部粒子的全局最优位置, 则对微环境网络中任意微环境, 成立: $fun(gBest_L) \geq fun(gBest_A) \geq fun(gBest)$. 因此, 类似式(6)、式(7)规定的粒子群优化算法中粒子速度与位置更新, 对任意一个微环境中的粒子群中的粒子使用式(9)、式(7)更新速度和位置, 该微环境中的粒子群中粒子位置收敛. 进一步, 对全部微环境中的粒子群, 虽然 $gBest$ 可能不出现在每个微环境的粒子群中, 但依据微环境网络中微环境的连接特性和式(8)可知, 由于任意两个微环境在微环境网络中有限步可达, 故全局最优位置 $gBest$ 可以经过有限次迭代传递到微环境网络中每个微环境上的粒子群(极端情况: 微环境网络中全部 m 个微环境线性排列, 设第一个微环境是A, 最后一个微环境是B. 若最优位置 $gBest$ 是A微环境粒子群中的局部最优位置, 则至多经过 $m-1$ 次迭代, B微环境上的局部最优位置优于 $m-1$ 次迭代前A节点的粒子群的局部最优位置, 即优于 $m-1$ 次迭代前的全局最优位置). 显然, 式(8)规定的的数据交换策略可以使全局最优位置 $gBest$ 在微环境网络中每个微环境上传播, 而全局最优位置 $gBest$ 可以传递到微环境网络中每个微环境, 意味着微环境网络中每个微环境上的粒子群本地局部历史位置 $gBest_L$ 距全局最优位置有限步可达, 表现为当算法6运行结束时, 每个微环境中粒子群的本地局部历史位置 $gBest_L$ 的适应度值差异较小.

使用PSO算法求解式(3)描述的电流迁移态序列片段 S 一元线性回归模型常系数 a 、 b 时, (a, b) 是PSO算法中的一个粒子. 使用MPSO算法求解电流迁移态序列片段 S 一元线性回归模型常系数的流程由算法7给出.

算法 7 基于MPSO算法的电流迁移态序列片段一元线性回归模型常系数求解算法

输入: 电流迁移态序列片段 S , 微环境网络拓扑结构 $meNet$, 适应度函数 fun , 特征维度 $varLen$, 粒子数 PN , 学习因子 c_1 、 c_2 , 惯性因子 ω , 最大迭代次数 $\max N$

输出: 电流迁移态序列片段 S 一元线性回归模型常系数 (a, b)

- 1) $m = \text{getNodeNum}(meNet)$; // 获取微环境网络中微环境的个数
- 2) 将 PN 个粒子分成 m 份, 每份中粒子的数量尽可能均匀, 记第 i 组粒子的数量为 m_i , $i=1,2,\dots,m$
- 3) 依据微环境网络拓扑结构 $meNet$ 生成每个微环境的邻居微环境列表, 记第 i 个微环境的邻居微环境列表为 $pNeighbor_i$, $i=1,2,\dots,m$
- 4) for $i=1$ to m

对第 i 个微环境,以本地粒子数 m_i 、粒子位置维度 $varLen$ 、学习因子 c_1 和 c_2 、惯性因子 ω 、最大迭代次数 $maxN$ 、适应度函数 fun 、邻居微环境节点列表 $pNeighbor_i$ 为输入参数调用算法6

5) 采集每个微环境中算法6的返回值 $gBest_L$ 的适应度值,将适应度值最小的记作 $gBest$

6) $(a, b) = gBest$

算法5、算法6和算法7使用适应度函数 fun 计算粒子的适应度值时,其输入为粒子的位置.算法7在计算粒子的适应度值时,还需要将电流迁移态序列片段 S 作为输入.使用算法7求解电流迁移态序列片段一元线性回归模型常系数的适应度函数依据式(4)设计,详细流程在算法8中给出.

算法8 计算电流迁移态序列片段一元线性回归模型常系数适应度值

输入:一元线性回归模型常系数 a, b , 电流迁移态序列片段 S

输出:一元线性回归模型常系数 a, b 的适应度值 $fValue$

1) $n = \text{length}(S)$

2) $fs = b, fValue = 0$; // fs 为一元回归模型对 S 第一个数据的估计

3) for $i = 1:n$

4) $tmp = S(i) - fs$

5) $fValue = fValue + tmp * tmp$

6) $fs = fs + a$; // fs 为一元回归模型对 S 下一个数据的估计

7) $fValue = fValue / 2/n$

3 实验结果与分析

实验使用的电流数据序列为智能建筑与建筑节能安徽省重点实验室自主研发的智能插座系统采集的微波炉工作时的电流观测数据,每2秒采集一次,共3 193个.使用算法1~算法4获取蕴含于电流数据序列中的电流迁移态序列,当 $\varepsilon = 0.01$ 、 $\Delta = 0.03$ 时,有184个电流迁移态序列.图4(a)给出了全部3 193个电流数据的序列以及184个电流迁移态序列的形态,图4(b)给出了前250个电流数据组成的电流数据序列片段以及蕴含其中的电流迁移态序列的形态.图4(a)和图4(b)中,电流数据序列使用蓝色线段表示,电流迁移态序列使用红色线段表示.显然,由于每个电流迁移态序列都是电流数据序列的片段,图4(a)和图4(b)中,每个红色线段都覆盖了同样的蓝色线段.同时,从图4(a)和图4(b)可以直观地发现:每个电流迁移态序列对应的都是电流数据序列中上升或者下降比较快的片段.

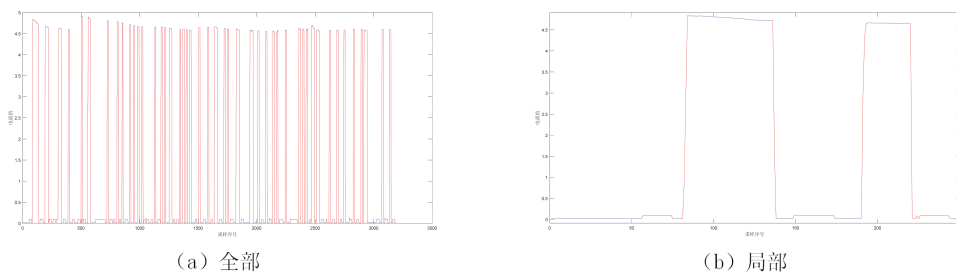


图4 电流数据序列与迁移态序列

实验分别使用算法5描述的PSO算法、文献[18]提出的混沌自适应粒子群优化算法(CAPSO算法)和算法6描述的基于MPSO算法求解式(3)规定的电流迁移态特征.对PSO算法,设置粒子数为400,学习因子 c_1 、 c_2 为2,惯性因子 ω 为0.8.对CAPSO算法,设置粒子数为400,惯性权值 w_{max} 和 w_{min} 分别为0.9和0.4,自适应加速度系数 c_{max} 和 c_{min} 分别为2.5和0.5,混沌搜索参数 ξ 为0.2.对MPSO算法,设置网络拓扑结构为图3所示的5行4列网格,共20个微环境节点;对每个微环境,粒子数设置为20(这样,全部微环境中的粒子数总和为400,与PSO算法和CAPSO算法一致),学习因子 c_1 、 c_2 为2,惯性因子 ω 为0.8.

表1和表2分别给出了重复20次使用PSO算法和CAPSO算法求解全部184个电流迁移态序列特征时,每次求解获得全部184个电流迁移态序列特征性能的最小值、最大值、平均值、中位值、下四分位Q1、上四分位Q3、方差以及使用时间等信息;使用MPSO算法构造全部184个电流迁移态序列特征时,每个微环境内求解出的全

部184个电流迁移态序列特征性能的最小值、最大值、平均值、中位值、下四分位Q1、上四分位Q3、方差以及使用时间等信息在表3中给出. 其最小值、最大值、平均值、中位值、下四分位Q1、上四分位Q3、方差是全部184个电流迁移态序列特征依据式(4)规定性能的统计值.

表 1 PSO算法的性能(迭代次数: 100)

	最小值	Q1	中位值	Q3	最大值	平均值	方差	时间/秒
1	$0.199\ 7\times 10^{-12}$	$0.293\ 1\times 10^{-10}$	$0.484\ 8\times 10^{-7}$	0.178 6	0.648	0.105 9	0.177 4	54.415 6
2	$0.024\ 2\times 10^{-12}$	$0.212\ 3\times 10^{-10}$	$0.493\ 4\times 10^{-7}$	0.178 6	0.648	0.105 9	0.177 4	53.436 6
3	$0.177\ 2\times 10^{-12}$	$0.442\ 7\times 10^{-10}$	$0.498\ 4\times 10^{-7}$	0.178 6	0.648	0.105 9	0.177 4	53.848 4
4	$0.112\ 1\times 10^{-12}$	$0.295\ 3\times 10^{-10}$	$0.493\ 1\times 10^{-7}$	0.178 6	0.648	0.105 9	0.177 4	53.687 8
5	$0.063\ 7\times 10^{-12}$	$0.225\ 5\times 10^{-10}$	$0.480\ 5\times 10^{-7}$	0.178 6	0.648	0.105 9	0.177 4	53.782 1
6	$0.158\ 3\times 10^{-12}$	$0.165\ 8\times 10^{-10}$	$0.483\ 2\times 10^{-7}$	0.178 6	0.648	0.105 9	0.177 4	54.063 9
7	$0.041\ 6\times 10^{-12}$	$0.183\ 4\times 10^{-10}$	$0.489\ 8\times 10^{-7}$	0.178 6	0.648	0.105 9	0.177 4	53.481 2
8	$0.060\ 7\times 10^{-12}$	$0.272\ 0\times 10^{-10}$	$0.519\ 7\times 10^{-7}$	0.178 6	0.648	0.105 9	0.177 4	54.248 2
9	$0.320\ 1\times 10^{-12}$	$0.297\ 3\times 10^{-10}$	$0.478\ 4\times 10^{-7}$	0.178 6	0.648	0.105 9	0.177 4	53.573 7
10	$0.039\ 2\times 10^{-12}$	$0.295\ 1\times 10^{-10}$	$0.488\ 5\times 10^{-7}$	0.178 6	0.648	0.105 9	0.177 4	53.589 7
11	$0.116\ 3\times 10^{-12}$	$0.192\ 4\times 10^{-10}$	$0.481\ 2\times 10^{-7}$	0.178 6	0.648	0.105 9	0.177 4	53.775 1
12	$0.107\ 3\times 10^{-12}$	$0.288\ 2\times 10^{-10}$	$0.478\ 5\times 10^{-7}$	0.178 6	0.648	0.105 9	0.177 4	53.721 2
13	$0.048\ 6\times 10^{-12}$	$0.170\ 3\times 10^{-10}$	$0.477\ 3\times 10^{-7}$	0.178 6	0.648	0.105 9	0.177 4	53.787 4
14	$0.015\ 5\times 10^{-12}$	$0.205\ 9\times 10^{-10}$	$0.482\ 8\times 10^{-7}$	0.178 6	0.648	0.105 9	0.177 4	53.984 5
15	$0.287\ 9\times 10^{-12}$	$0.276\ 0\times 10^{-10}$	$0.497\ 0\times 10^{-7}$	0.178 6	0.648	0.105 9	0.177 4	53.827 0
16	$0.107\ 5\times 10^{-12}$	$0.198\ 1\times 10^{-10}$	$0.504\ 5\times 10^{-7}$	0.178 6	0.648	0.105 9	0.177 4	53.360 9
17	$0.097\ 4\times 10^{-12}$	$0.264\ 3\times 10^{-10}$	$0.475\ 5\times 10^{-7}$	0.178 6	0.648	0.105 9	0.177 4	53.669 4
18	$0.065\ 2\times 10^{-12}$	$0.337\ 1\times 10^{-10}$	$0.504\ 6\times 10^{-7}$	0.178 6	0.648	0.105 9	0.177 4	53.787 6
19	$0.079\ 9\times 10^{-12}$	$0.178\ 6\times 10^{-10}$	$0.492\ 5\times 10^{-7}$	0.178 6	0.648	0.105 9	0.177 4	53.239 1
20	$0.030\ 4\times 10^{-12}$	$0.317\ 7\times 10^{-10}$	$0.479\ 2\times 10^{-7}$	0.178 6	0.648	0.105 9	0.177 4	53.903 5

表 2 CAPSO算法的性能(迭代次数: 100)

	最小值	Q1	中位值	Q3	最大值	平均值	方差	时间/秒
1	$0.000\ 4\times 10^{-8}$	$0.153\ 2\times 10^{-5}$	$0.267\ 6\times 10^{-4}$	0.178 6	0.648 0	0.105 9	0.177 4	70.073 6
2	$0.082\ 7\times 10^{-8}$	$0.103\ 3\times 10^{-5}$	$0.155\ 3\times 10^{-4}$	0.178 6	0.648 0	0.105 9	0.177 4	69.879 0
3	$0.141\ 6\times 10^{-8}$	$0.148\ 7\times 10^{-5}$	$0.174\ 8\times 10^{-4}$	0.178 6	0.648 0	0.105 9	0.177 4	70.667 2
4	$0.234\ 3\times 10^{-8}$	$0.102\ 2\times 10^{-5}$	$0.243\ 0\times 10^{-4}$	0.178 6	0.648 0	0.105 9	0.177 4	70.796 0
5	$0.171\ 9\times 10^{-8}$	$0.082\ 9\times 10^{-5}$	$0.145\ 3\times 10^{-4}$	0.178 6	0.648 0	0.105 9	0.177 4	69.779 0
6	$0.452\ 0\times 10^{-13}$	$0.118\ 3\times 10^{-5}$	$0.227\ 1\times 10^{-4}$	0.178 6	0.648 0	0.105 9	0.177 4	69.999 0
7	$0.103\ 9\times 10^{-8}$	$0.079\ 6\times 10^{-5}$	$0.175\ 6\times 10^{-4}$	0.178 6	0.648 4	0.105 9	0.177 4	70.552 7
8	$0.062\ 7\times 10^{-8}$	$0.086\ 1\times 10^{-5}$	$0.141\ 3\times 10^{-4}$	0.178 6	0.649 0	0.105 9	0.177 5	70.319 2
9	$0.070\ 3\times 10^{-8}$	$0.082\ 7\times 10^{-5}$	$0.190\ 3\times 10^{-4}$	0.178 6	0.648 0	0.105 9	0.177 4	70.439 7
10	$0.062\ 1\times 10^{-8}$	$0.092\ 6\times 10^{-5}$	$0.227\ 9\times 10^{-4}$	0.178 6	0.648 0	0.105 9	0.177 4	70.772 4
11	$0.293\ 8\times 10^{-8}$	$0.081\ 2\times 10^{-5}$	$0.191\ 9\times 10^{-4}$	0.178 6	0.648 1	0.105 9	0.177 4	70.279 3
12	$0.230\ 0\times 10^{-8}$	$0.079\ 2\times 10^{-5}$	$0.193\ 9\times 10^{-4}$	0.178 6	0.648 1	0.105 9	0.177 4	70.090 7
13	$0.179\ 1\times 10^{-8}$	$0.146\ 1\times 10^{-5}$	$0.226\ 5\times 10^{-4}$	0.178 6	0.648 1	0.105 9	0.177 4	70.148 1
14	$0.298\ 7\times 10^{-8}$	$0.083\ 3\times 10^{-5}$	$0.207\ 9\times 10^{-4}$	0.178 6	0.648 0	0.105 9	0.177 4	70.495 0
15	$0.068\ 6\times 10^{-8}$	$0.125\ 4\times 10^{-5}$	$0.203\ 2\times 10^{-4}$	0.178 6	0.648 0	0.105 9	0.177 4	70.257 5
16	$0.083\ 4\times 10^{-8}$	$0.104\ 7\times 10^{-5}$	$0.182\ 6\times 10^{-4}$	0.178 6	0.648 0	0.105 9	0.177 4	69.889 7
17	$0.232\ 1\times 10^{-8}$	$0.146\ 9\times 10^{-5}$	$0.310\ 2\times 10^{-4}$	0.178 6	0.648 1	0.105 9	0.177 4	69.535 7
18	$0.102\ 7\times 10^{-8}$	$0.075\ 8\times 10^{-5}$	$0.209\ 4\times 10^{-4}$	0.178 6	0.648 4	0.105 9	0.177 4	69.479 5
19	$0.004\ 6\times 10^{-8}$	$0.078\ 4\times 10^{-5}$	$0.187\ 7\times 10^{-4}$	0.178 6	0.648 0	0.105 9	0.177 4	69.391 1
20	$0.026\ 3\times 10^{-8}$	$0.125\ 6\times 10^{-5}$	$0.142\ 5\times 10^{-4}$	0.178 6	0.648 1	0.105 9	0.177 4	69.437 7

表1和表2中,时间栏记录的是PSO算法和CAPSO算法求解全部184个电流迁移态序列特征使用的时间,而表3中时间栏记录的为MPSO算法运行一次的时间.文献[19]对分布式平台的通信开销进行研究,发现基于局域网的分布式平台任意两个节点间的平均延迟不大于0.27毫秒,网络性能可满足对通信时间的要求.

表3 MPSO算法的性能(迭代次数:100)

	最小值	Q1	中位值	Q3	最大值	平均值	方差	时间/秒
1	0.0218×10^{-9}	0.0734×10^{-7}	0.0796×10^{-5}	0.178 6	0.648	0.105 9	0.177 4	
2	0.0104×10^{-9}	0.0921×10^{-7}	0.2225×10^{-5}	0.178 6	0.648	0.105 9	0.177 4	
3	0.0117×10^{-9}	0.0675×10^{-7}	0.3116×10^{-5}	0.178 6	0.648	0.105 9	0.177 4	
4	0.0054×10^{-9}	0.0945×10^{-7}	0.2877×10^{-5}	0.178 6	0.648	0.105 9	0.177 4	
5	0.1296×10^{-9}	0.1074×10^{-7}	0.1562×10^{-5}	0.178 6	0.648	0.105 9	0.177 4	
6	0.0046×10^{-9}	0.0622×10^{-7}	0.1857×10^{-5}	0.178 6	0.648	0.105 9	0.177 4	
7	0.0125×10^{-9}	0.1372×10^{-7}	0.2376×10^{-5}	0.178 6	0.648	0.105 9	0.177 4	
8	0.0138×10^{-9}	0.0922×10^{-7}	0.3155×10^{-5}	0.178 6	0.648	0.105 9	0.177 4	
9	0.0116×10^{-9}	0.1547×10^{-7}	0.2132×10^{-5}	0.178 6	0.648	0.105 9	0.177 4	
10	0.0312×10^{-9}	0.0710×10^{-7}	0.1634×10^{-5}	0.178 6	0.648	0.105 9	0.177 4	共耗时
11	0.0021×10^{-9}	0.0816×10^{-7}	0.3047×10^{-5}	0.178 6	0.648	0.105 9	0.177 4	261.867 1
12	0.0122×10^{-9}	0.0662×10^{-7}	0.2279×10^{-5}	0.178 6	0.648	0.105 9	0.177 4	
13	0.0089×10^{-9}	0.1012×10^{-7}	0.1932×10^{-5}	0.178 6	0.648	0.105 9	0.177 4	
14	0.0010×10^{-9}	0.0941×10^{-7}	0.1417×10^{-5}	0.178 6	0.648	0.105 9	0.177 4	
15	0.0074×10^{-9}	0.0970×10^{-7}	0.1592×10^{-5}	0.178 6	0.648	0.105 9	0.177 4	
16	0.0156×10^{-9}	0.1106×10^{-7}	0.1689×10^{-5}	0.178 6	0.648	0.105 9	0.177 4	
17	0.0019×10^{-9}	0.1169×10^{-7}	0.1032×10^{-5}	0.178 6	0.648	0.105 9	0.177 4	
18	0.0273×10^{-9}	0.1003×10^{-7}	0.1677×10^{-5}	0.178 6	0.648	0.105 9	0.177 4	
19	0.0191×10^{-9}	0.0663×10^{-7}	0.2471×10^{-5}	0.178 6	0.648	0.105 9	0.177 4	
20	0.0106×10^{-9}	0.1176×10^{-7}	0.2205×10^{-5}	0.178 6	0.648	0.105 9	0.177 4	

对比表1、表2和表3中最小值、最大值、平均值、中位值、Q1、Q3、方差等数据,可以断言,无论是PSO算法、CAPSO算法还是MPSO算法都可以用于电流迁移态序列特征的有效求解.考虑到PSO算法和CAPSO算法求解最优化问题时一次执行获得的最优解只是近似最优解,为获得更接近最优解的近似最优解,一般需要重复多次执行算法,这使得PSO算法和CAPSO算法求解最优化问题时需要更多的时间.使用MPSO算法求解最优化问题时,算法的1次执行可同时获得与微环境个数相同数量的近似最优解,效率上要比使用PSO算法高许多.由表1和表2中时间数据可知,每次PSO算法求解全部184个电流迁移态序列的特征使用时间的最小值为53.239 1秒、最大值为54.415 6秒,平均值为53.759 2秒,方差为0.284 5秒,全部20次求解共耗时1 075.183 1秒;每次CAPSO算法求解全部184个电流迁移态序列特征使用时间的最小值为69.391 1秒、最大值为70.796 0秒,平均值为70.114 1秒,方差为0.440 7秒,全部20次求解共耗时1 402.281 8秒;而表3中时间数据显示,MPSO算法一次运行即可获得20组近似最优解,共耗时261.867 1秒,平均每个近似最优解的求解耗时13.093 4秒,远小于一次用PSO算法和CAPSO算法求解全部184个电流迁移态序列特征平均需要的时间,MPSO算法用时仅是PSO算法和CAPSO算法的24.355 6%和18.674 4%.

图5、图6分别给出了使用PSO算法和CAPSO算法一次求解全部184个电流迁移态序列特征时性能的最小值、最大值、平均值、中位值、Q1以及Q3随着迭代次数增加的变化情况.显然,随着迭代次数的增加,性能的最小值、最大值、平均值、中位值、Q1以及Q3快速下降后渐趋稳定.图7给出了使用MPSO算法求解全部184个电流迁移态序列特征时一个微环境内性能的最小值、最大值、平均值、中位值、Q1以及Q3随着迭代次数增加的变化情况.显然,与使用PSO算法一次求解全部184个电流迁移态序列特征时性能的变化一样,随着迭代次数的增加,MPSO算法性能的最小值、最大值、平均值、中位值、Q1以及Q3快速下降后渐趋稳定.对照图5、图6、图7,CAPSO算法在迭代50次后,性能的最小值、最大值、平均值、中位值、Q1以及Q3都渐趋稳定;而PSO算法和MPSO算法在迭代20次后,性能的最小值、最大值、平均值、中位值、Q1以及Q3就渐趋稳定.

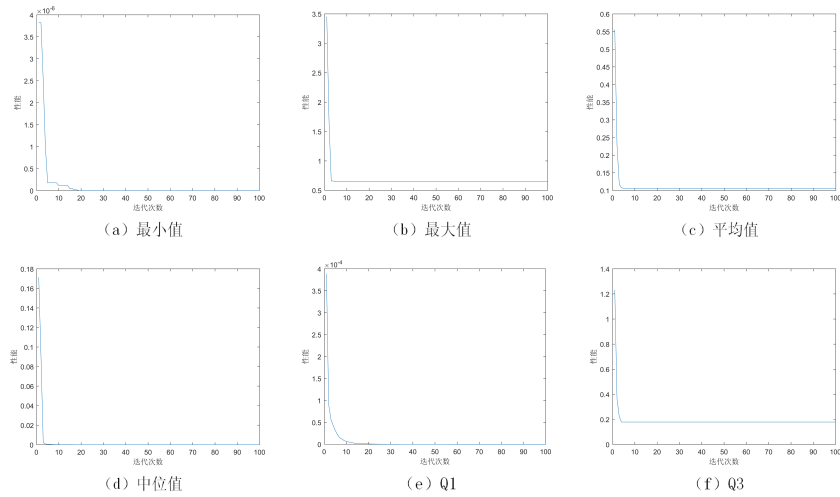


图 5 PSO算法的性能(迭代次数: 100)

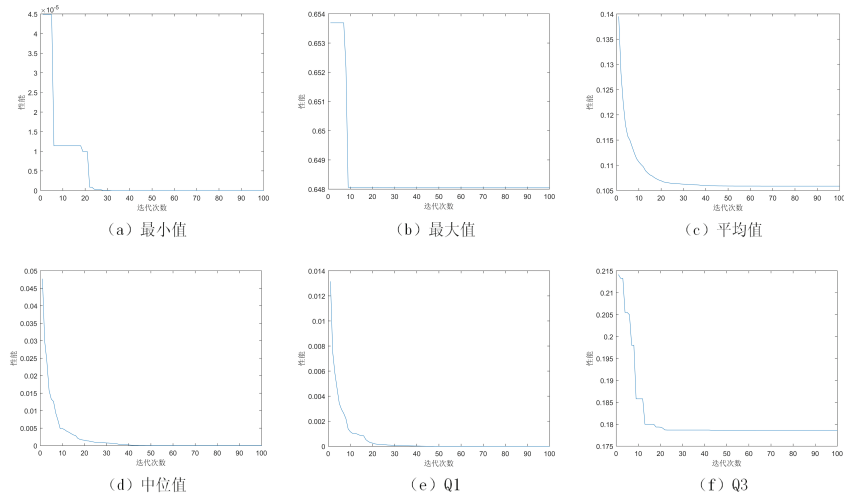


图 6 CAPSO算法的性能(迭代次数: 100)

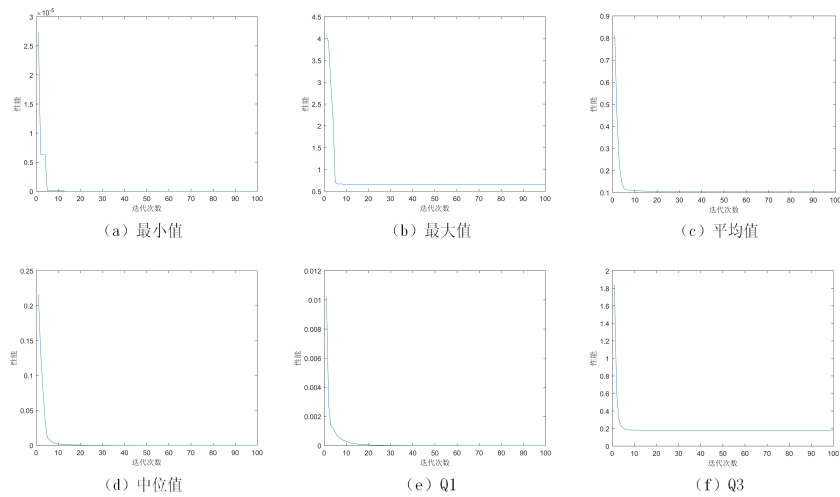


图 7 MPSO算法的性能(迭代次数: 100)

图8给出了迭代次数为20时,使用PSO算法一次求解全部184个电流迁移态序列特征性能的最小值、最大值、平均值、中位值、Q1以及Q3随着迭代次数增加的变化情况.显然,随着迭代次数的增加,性能的最小值、最大值、平均值、中位值、Q1以及Q3快速下降,在接近迭代后渐趋稳定,这意味着使用PSO算法求解电流迁移态序

列特征时, 迭代次数设置为20是合适的. 重复20次使用PSO算法求解全部184个电流迁移态序列特征, 使用时间的分布直方图如图9所示. PSO算法使用时间的最小值为11.265 1秒、最大值为11.821 6秒, 平均值为11.405 6秒, 方差为0.148 7秒, 显然, 使用PSO算法求解电流迁移态序列特征是稳定的.

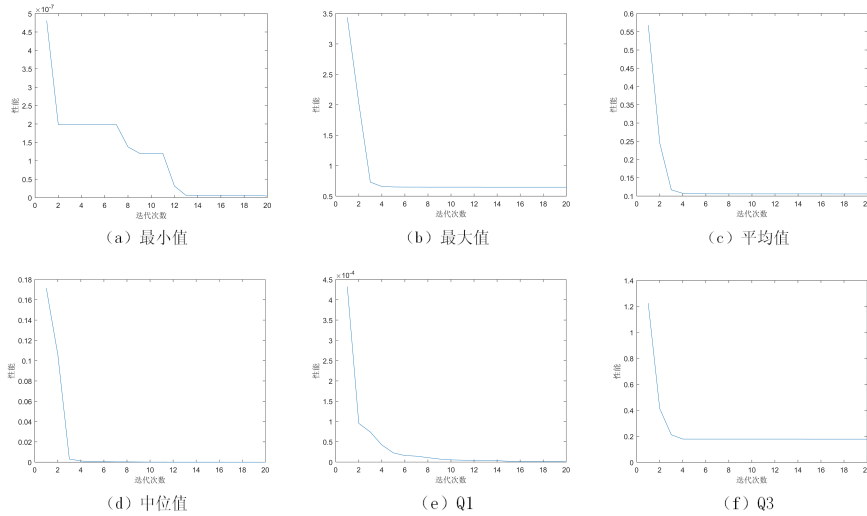


图 8 PSO算法的性能 (迭代次数: 20)

图10给出了迭代次数为20时, 使用MPSO算法求解全部184个电流迁移态序列特征时, 一个微环境中算法性能的最小值、最大值、平均值、中位值、Q1以及Q3随着迭代次数增加的变化情况. 显然, 随着迭代次数的增加, 性能的最小值、最大值、平均值、中位值、Q1以及Q3快速下降, 在接近迭代后渐趋稳定, 这意味着使用MPSO算法求解电流迁移态序列特征时, 迭代次数设置为20是合适的. 实验中, 迭代次数设置为20时, 20个微种群完成184个电流迁移态序列特征的求解用时56.323 9秒, 每个微环境平均用时2.816 2秒, 是使用PSO算法用时的24.691 4%, 这再次印证了使用MPSO算法可以快速地完成电流迁移态序列特征的求解.

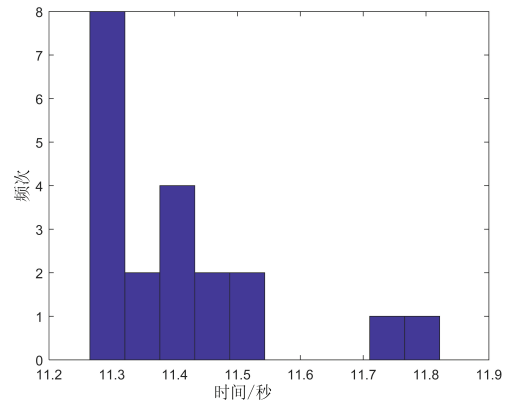


图 9 PSO算法用时分布直方图 (迭代次数: 20)

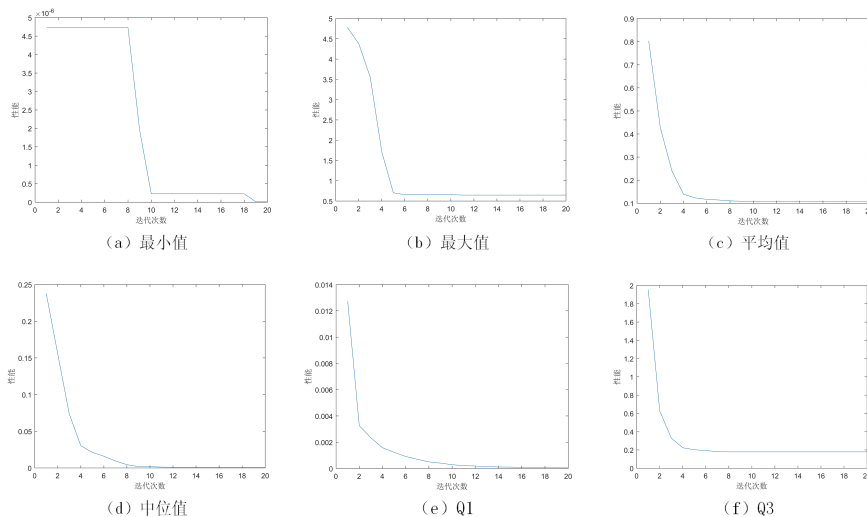


图 10 MPSO算法的性能 (迭代次数: 20)

使用BP神经网络对提取出的电流迁移态特征进行电器状态的辨识,重复进行实验100次,图11给出了100次实验电器状态辨识准确率(Accuracy)的分布情况:最大值为1,最小值为0.781 8,平均值为0.979 3,中位值为0.981 8.可知使用本文所提电流迁移态序列特征模型提取出的特征能够以较高的稳定性和准确率实现对电器状态的辨识.

实验使用曙光W760-g20服务器完成,该服务器的CPU为2个Intel(R) Xeon(R) CPU E5-2637 v3,192 GB ECC内存,操作系统为Windows 10,Matlab使用R2017a版本.

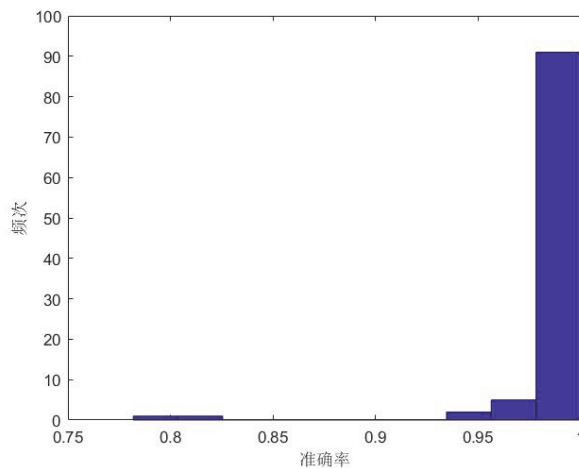


图 11 电器状态辨识准确率分布直方图

4 结论与展望

物联网技术的进步与普及丰富了用户用电态势在线监测的手段,促进了需求侧用电过程的监测与管理.特别是针对诸如安全用电、智慧园区、智慧社区、智慧建筑等场景的应用需求,其中电器及其工作模式的在线辨识为越来越多的研究者所关注.为快速辨识电器的工作状态,本文使用一元回归模型描述电流迁移态序列的特征,通过电器工作状态切换的辨识进行电器工作模式的在线辨识.电流迁移态序列的一元回归特征计算量小的优势,保证了电器工作模式在线辨识的及时性.同时,为提升电器工作模式在线辨识的准确性,面向电流迁移态序列特征的优化,本文还提出了基于微环境的粒子群优化算法(MPSO算法).与使用PSO算法和CAPSO算法优化电流迁移态序列特征相比,MPSO算法通过多个微环境的协作,使用相比PSO算法和CAPSO算法更少的粒子数进行计算,在取得与PSO算法和CAPSO算法一致性能的同时,执行速度显著提升.

微环境网络是MPSO算法的计算架构,本文使用了只有邻居微环境才能够交换信息的 5×4 网格实现了MPSO算法,并高效地优化了电流迁移态序列的一元回归特征.网络拓扑对MPSO算法性能的影响值得进一步研究.

本文所提MPSO算法本质上是该算法在Matlab环境中的串行实现.诸如边缘计算、群智能建筑平台等真实的分布式信息处理环境中,MPSO算法计算过程的优化以及网络通信开销,对算法性能的影响是需要关注的.进一步,不同微环境信息处理的同步与异步对算法性能的影响、深度算力以及异构算力下的MPSO算法及其应用都是值得深入开展的工作.

参考文献:

- [1] 王鹏,林佳颖,郭岫,等.配用电数据分析及应用[J].电网技术,2017,41(10):3333-3340.
WANG P, LIN J Y, GUO S, et al. Distribution system data analytics and applications[J]. Power System Technology, 2017, 41(10): 3333-3340. (in Chinese)
- [2] 周勇军,吴元香,董智华,等.基于模体挖掘与调和函数半监督学习的非侵入式负荷监测[J].电力自动化设备,2022,42(7):3-10.
ZHOU Y J, WU Y X, DONG Z H, et al. Non-intrusive load monitoring based on motif mining and harmonic function based semi-supervised learning[J]. Electric Power Automation Equipment, 2022, 42(7): 3-10. (in Chinese)

- [3] 朱天怡, 艾芊, 贺兴, 等. 基于数据驱动的用电行为分析方法及应用综述[J]. 电网技术, 2020, 44(9): 3497-3507.
ZHU T Y, AI Q, HE X, et al. An overview of data-driven electricity consumption behavior analysis method and application[J]. Power System Technology, 2020, 44(9): 3497-3507. (in Chinese)
- [4] LIN Y H, TSAI M S. Development of an improved time-frequency analysis-based nonintrusive load monitor for load demand identification[J]. IEEE Transactions on Instrumentation and Measurement, 2014, 63(6): 1470-1483.
- [5] SAITOH T, OSAKI T, KONISHI R, et al. Current sensor based home appliance and state of appliance recognition[J]. SICE Journal of Control, Measurement, and System Integration, 2010, 3(2): 86-93.
- [6] YAN D, JIN Y, SUN H S, et al. Household appliance recognition through a Bayes classification model[J]. Sustainable Cities and Society, 2019, 46: 101393.
- [7] MULINARI B M, DE CAMPOS D P, DA COSTA C H, et al. A new set of steady-state and transient features for power signature analysis based on V-I trajectory[C]//2019 IEEE PES Innovative Smart Grid Technologies Conference-Latin America (ISGT Latin America). Gramado, Brazil. IEEE, 2019: 1-6.
- [8] CHEN J F, WANG X, ZHANG X T, et al. Temporal and spectral feature learning with two-stream convolutional neural networks for appliance recognition in NILM[J]. IEEE Transactions on Smart Grid, 2022, 13(1): 762-772.
- [9] NORFORD L K, LEEB S B. Non-intrusive electrical load monitoring in commercial buildings based on steady-state and transient load-detection algorithms[J]. Energy and Buildings, 1996, 24(1): 51-64.
- [10] CHANG H H. Non-intrusive demand monitoring and load identification for energy management systems based on transient feature analyses[J]. Energies, 2012, 5(11): 4569-4589.
- [11] LE T T H, KANG H, KIM H. Household appliance classification using lower odd-numbered harmonics and the bagging decision tree[J]. IEEE Access, 2020, 8: 55937-55952.
- [12] BOUHOUS A S, GKAITATZIS P A, PANAGIOTOU E, et al. A NILM algorithm with enhanced disaggregation scheme under harmonic current vectors[J]. Energy and Buildings, 2019, 183: 392-407.
- [13] HIMEUR Y, ALSALEMI A, BENSALI F, et al. Robust event-based non-intrusive appliance recognition using multi-scale wavelet packet tree and ensemble bagging tree[J]. Applied Energy, 2020, 267: 114877.
- [14] GUO L Y, WANG S X, CHEN H W, et al. A load identification method based on active deep learning and discrete wavelet transform[J]. IEEE Access, 2020, 8: 113932-113942.
- [15] JIMENEZ Y, DUARTE C, PETIT J, et al. Feature extraction for nonintrusive load monitoring based on S-Transform[C]//2014 Clemson University Power Systems Conference. Clemson, SC, USA. IEEE, 2014: 1-5.
- [16] 张广金. 一种面向用电设备管理的智能插座系统设计与实现[D]. 合肥: 安徽建筑大学, 2017.
ZHANG G J. Design and implementation of smart socket system for electricity equipment management[D]. Hefei: Anhui Jianzhu University, 2017. (in Chinese)
- [17] 方潜生, 张广金, 张振亚, 等. 一种智能型插座: CN205039355U[P]. 2016-02-17.
FANG Q S, ZHANG G J, ZHANG Z Y, et al. An intelligent socket: CN205039355U[P]. 2016-02-17. (in Chinese)
- [18] DUAN Y X, CHEN N, CHANG L J, et al. CAPSO: Chaos adaptive particle swarm optimization algorithm[J]. IEEE Access, 2022, 10: 29393-29405.
- [19] ZHANG Z Y, FANG B, WANG P, et al. A local area network-based insect intelligent building platform[J]. International Journal of Pattern Recognition and Artificial Intelligence, 2023, 37(2): 2359004.

责任编辑: 张自强