FUTURE COMPUTING 2021

The Thirteenth International Conference on Future Computational Technologies and Applications

ISBN: 978-1-61208-846-4

April 18 - 22, 2021

FUTURE COMPUTING 2021 Editors

Hiroyuki Sato, The University of Tokyo, Japan

FUTURE COMPUTING 2021

Forward

The Thirteenth International Conference on Future Computational Technologies and Applications (FUTURE COMPUTING 2021), held on April 18 - 22, 2021, continued a series of events targeting advanced computational paradigms and their applications. The target was to cover (i) the advanced research on computational techniques that apply the newest human-like decisions, and (ii) applications on various domains. The new development led to special computational facets on mechanism-oriented computing, large-scale computing and technology-oriented computing. They are largely expected to play an important role in cloud systems, on-demand services, autonomic systems, and pervasive applications and services.

The conference had the following tracks:

- Computing technologies
- Computational intelligence strategies
- Challenges

Similar to the previous edition, this event attracted excellent contributions and active participation from all over the world. We were very pleased to receive top quality contributions.

We take here the opportunity to warmly thank all the members of the FUTURE COMPUTING 2021 technical program committee, as well as the numerous reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors that dedicated much of their time and effort to contribute to FUTURE COMPUTING 2021.

Also, this event could not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the FUTURE COMPUTING 2021 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope FUTURE COMPUTING 2021 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the area of future computational technologies and applications..

FUTURE COMPUTING 2021 General Chair

Sandra Sendra, Universitat Politecnica de Valencia, Universidad de Granada, Spain

FUTURE COMPUTING 2021 Steering Committee

Hiroyuki Sato, The University of Tokyo, Japan
Sergio Ilarri, University of Zaragoza, Spain
Jay Lofstead, Sandia National Laboratories, USA

FUTURE COMPUTING 2021 Publicity Chair

Jose Luis García, Universitat Politecnica de Valencia, Spain
Lorena Parra, Universitat Politecnica de Valencia, Spain

FUTURE COMPUTING 2021 Industry/Research Advisory Committee

Francesc Guim, Intel Corporation, Spain
Yasushi Kambayashi, Nippon Institute of Technology, Japan
Kendall E. Nygard, North Dakota State University - Fargo, USA
Alex Wijesinha, Towson University, USA
Albert Zomaya, University of Sydney, Australia
Christos J. Bouras, University of Patras, Greece

FUTURE COMPUTING 2021

Committee

FUTURE COMPUTING 2021 General Chair

Sandra Sendra, Universitat Politecnica de Valencia, Universidad de Granada, Spain

FUTURE COMPUTING 2021 Steering Committee

Hiroyuki Sato, The University of Tokyo, Japan
Sergio Ilarri, University of Zaragoza, Spain
Jay Lofstead, Sandia National Laboratories, USA

FUTURE COMPUTING 2021 Industry/Research Advisory Committee

Francesc Guim, Intel Corporation, Spain
Yasushi Kambayashi, Nippon Institute of Technology, Japan
Kendall E. Nygard, North Dakota State University - Fargo, USA
Alex Wijesinha, Towson University, USA
Albert Zomaya, University of Sydney, Australia
Christos J. Bouras, University of Patras, Greece

FUTURE COMPUTING 2021 Publicity Chairs

Jose Luis García, Universitat Politecnica de Valencia, Spain
Lorena Parra, Universitat Politecnica de Valencia, Spain

FUTURE COMPUTING 2021 Technical Program Committee

Andrew Adamatzky, University of the West of England, Bristol, UK
Ehsan Atoofian, Lakehead University, Canada
Yadu Babuji, University of Chicago, USA
Bernhard Bandow, GWDG, Göttingen, Germany
Kaustav Basu, The Laboratory for Networked Existence (NetXT), USA
Rudolf Berrendorf, Bonn-Rhein-Sieg University, Germany
Christos J. Bouras, University of Patras, Greece
Massimiliano Caramia, University of Rome "Tor Vergata", Italy
Ryan Chard, Argonne National Laboratory, USA
Nan-Yow Chen, National Center for High-Performance Computing (NCHC), Taiwan
Sunil Choenni, Ministry of Justice and Security / Rotterdam University of Applied Sciences, Netherlands
Fabio D'Andreagiovanni, CNRS & UTC - Sorbonne, France
Leandro Dias da Silva, Universidade Federal de Alagoas, Brazil
Félix J. García Clemente, University of Murcia, Spain

Apostolos Gkamas, University Ecclesiastical Academy of Vella of Ioannina, Greece
Victor Govindaswamy, Concordia University - Chicago, USA
Francesc Guim, Intel Corporation, Spain
Tzung-Pei Hong, National University of Kaohsiung, Taiwan
Wei-Chiang Hong, School of Computer Science and Technology - Jiangsu Normal University, China
Sergio Ilarri, University of Zaragoza, Spain
Yuji Iwahori, Chubu University, Japan
Yasushi Kambayashi, Nippon Institute of Technology, Japan
Mehdi Kargar, Ted Rogers School of Management - Ryerson University, Toronto, Canada
Michihiro Koibuchi, National Institute of Informatics, Japan
Zbigniew Kokosinski, Cracow University of Technology, Poland
Carlos León-de-Mora, Universidad de Sevilla, Spain
Jay Lofstead, Sandia National Laboratories, USA
Carlos M. Travieso-González, University of Las Palmas de Gran Canaria, Spain
Giuseppe Mangioni, DIEEI - University of Catania, Italy
Wail Mardini, Jordan University of Science and Technology, Jordan
Isabel Muench, German Federal Office for Information Security, Germany
Anand Nayyar, Duy Tan University, Vietnam
Kendall E. Nygard, North Dakota State University - Fargo, USA
Carla Osthoff, National Laboratory for Scientific Computing, Brazil
Fred Petry, Naval Research Laboratory, USA
Wajid Rafique, Nanjing University, China
Eric Renault, Télécom SudParis | Institut Polytechnique de Paris, France
Carsten Röcker, FraunhoferIOSB-INA, Germany
Hiroyuki Sato, The University of Tokyo, Japan
Andrew Schumann, University of Information Technology and Management in Rzeszow, Poland
Friedhelm Schwenker, Ulm University, Germany
Zbigniew Suraj, University of Rzeszów, Poland
Teng Wang, Oracle, USA
Alex Wijesinha, Towson University, USA
Hongji Yang, Leicester University, UK
Peng-Yeng Yin, National Chi Nan University, Taiwan
Aleš Zamuda, University of Maribor, Slovenia
Claudio Zandron, University of Milano-Bicocca, Milan, Italy
Minjia Zhang, Microsoft AI and Research, USA
Albert Zomaya, University of Sydney, Australia

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

FOSDA: A Hybrid Disaggregated HPC Architecture based on Distributed Nanoseconds Optical Switches

Xiaotao Guo
Institute of Photonic Integration
Eindhoven University of Technology
Eindhoven, the Netherlands
e-mail: x.guo@tue.nl

Xuwei Xue
Institute of Photonic Integration
Eindhoven University of Technology
Eindhoven, the Netherlands
e-mail: x.xue.1@tue.nl

Bitao Pan
Institute of Photonic Integration
Eindhoven University of Technology
Eindhoven, the Netherlands
e-mail: b.pan@tue.nl

Fulong Yan
Institute of Photonic Integration
Eindhoven University of Technology
Eindhoven, the Netherlands
e-mail: f.yan@tue.nl

Georgios Exarchakos
Institute of Photonic Integration
Eindhoven University of Technology
Eindhoven, the Netherlands
e-mail: G.Exarchakos@tue.nl

Nicola Calabretta
Institute of Photonic Integration
Eindhoven University of Technology
Eindhoven, the Netherlands
e-mail: N.Calabretta@tue.nl

Abstract*— Aiming at solving the issues of low resource utilization and high operational cost in current node-centric High Performance Computing (HPC) architecture, we present and investigate a novel hybrid disaggregated HPC architecture based on nano-seconds fast optical switch (FOSDA). The fast optical switch connects Central Processing Unit (CPU) and memory nodes for the communication of high bandwidth and low latency, while storage nodes are interconnected by electrical packet switches. The performance of the FOSDA in terms of workload acceptance rate, resource utilization, power consumption and capital/operational cost is numerically assessed and compared with current node-centric HPC architectures. Compared with a node-centric HPC architecture of 320 nodes, FOSDA performs 36.6% higher CPU, and 21.5% higher memory resource utilization with 45.5% less active hardware resource, as well as 46.8% less power consumption. When scaling the HPC network to 2304 nodes, FOSDA also achieves 33.6% higher CPU and 48.5% higher memory utilization while saving 50.4% power consumption. In the cost analysis, FOSDA decreases operational cost by 46.7% with only 19.8% more capital cost.

Keywords- HPC network; disaggregated architecture; fast optical switch.

I. INTRODUCTION

Facing the rapidly increasing diversity and scale of big data computing applications, HPC system has evolved from Symmetric Multi-Processing (SMP) architecture into cluster architecture consisting of thousands of computing nodes. With a tenfold performance increase every four years [1], the world's fastest HPC architecture Summit supports up to 4608 nodes [2]. All the hardware resources (i.e., CPU, memory, and network) are closely coupled in the computing nodes of the current HPC system (named node-centric architecture). Yet, various scientific computing applications have different resource requirements [3]. Memory intensive applications require sufficient memory resource to realize the in-memory and parallel data processing, while network intensive applications rely on network resources to implement dense inter-node communication. The tethered resources in computing nodes may lead to waste of specific resources when other resources are mostly used. These wasted resources also consume a large amount of power and operational cost. In addition, node-centric architecture also increases the upgrade cost. The whole node needs to be upgraded when only a specific hardware component comes to the end of the lifecycle. Therefore, it is necessary to develop a novel HPC architecture to provide more flexible resource provision, better performance for various applications, and more cost-efficient maintaining.

Recently, a promising disaggregated architecture is proposed for solving issues of low resource utilization, high power consumption, and high operation cost in current node-centric architectures [4]-[6]. In the disaggregated architecture, the on-board data bus connecting hardware in each computing node is replaced by the network interconnection. There have been several studies to implement the disaggregated architecture. The Rack Scale Design (RSD) from Intel sets up the independent storage resource management system [7], but CPU and memory are still fixed in the computing node. A disaggregated memory system is emulated based on the Xen hypervisor [8], while a remote memory paging system was designed based on the (Remote Direct Memory Access) RDMA protocol [9]. Meanwhile, a distributed operating system was also developed for the disaggregated architecture [10]. These solutions are based on current multi-tier electrical networks for thousands of nodes interconnection. However, the multistage switching in electrical switch operating at a high data rate results in high cost, O/E power consumption, and node-to-node latency. Some networks like Dragonfly [11] can reduce the latency and cost of the HPC network based on high radix switches, while having disadvantages of limited bandwidth and resiliency. Some efforts seek the optical network of high bandwidth and low latency for the disaggregated architecture. Compared with current electrical networks, the optical switching technologies are transparent

ISBN: 978-1-61208-846-4

to the data rate and packet protocol. Field-Programmable Gate Array (FPGA) based programmable Network Interface Card (NIC) and Optical Circuit Switch (OCS) of large radix and high data rate were developed in [12][13] for the hardware nodes interconnection. Based on hybrid OCS and electrical switches, a disaggregated architecture “dReDBox” was proposed in [14]. However, it has been demonstrated in [15] that the network latency has more impact to the performance of the disaggregated network than the bandwidth, and sub-microsecond network latency is necessary for the hardware nodes communication. The milliseconds reconfiguration of OCS may degrade the performance and flexible resource provision of the disaggregated architecture. To reduce the network switching time, an Optical Packet Switch (OPS) named Hipoλaos was proposed based on the tunable wavelength converter [16], but this structure could have some practical implementation issues such as high-speed operation and format-transparent operation. Therefore, a scalable interconnection network of high bandwidth and low latency for the disaggregated HPC system remains an open research problem to be solved to provide flexible resource provision and reduce operational cost.

In this work, we present a novel disaggregated architecture FOSDA based on distributed nanoseconds Fast Optical Switches (FOS) for the HPC system. Instead of multi-tier electrical networks in the current node-centric architecture, a novel flat optical interconnect network of high bandwidth is applied in the FOSDA. Exploiting the optical switch of data rate and packet protocol transparence, the FOSDA can provide a data rate of up to hundreds of gigabits for hardware communication. Meanwhile, considering the low latency requirement of the disaggregated architecture, the FOSDA is designed exploiting Semiconductor Optical Amplifier (SOA) based optical switching technologies. Benefiting from nanoseconds switching time, hardware nodes in the FOSDA are interconnected with an interconnect network of low latency and fast reconfiguration, and packet contentions are solved by the optical flow control protocol without optical buffer. Based on the distributed structure of the FOS and parallel processing for each channel, the FOSDA is scalable to interconnect thousands of hardware nodes. The FOSDA performance is assessed in terms of workload acceptance rate, resource utilization and power consumption. The evaluation results show that the FOSDA achieves a 13% higher acceptance rate, up to 36.6% higher CPU, and 21.5% higher memory resource utilization, as well as 46.8% less power consumption, compared with node-centric HPC architectures. With a large network scale of 2304 nodes, FOSDA performs 33.6% higher CPU utilization, 48.5% higher memory utilization, and 50.4% less power consumption. In addition, the capital and operational cost of FOSDA are also investigated and compared with node-centric HPC architectures. It is demonstrated that FOSDA requires 46.7% less operational cost with only 19.8% more capital cost.

The paper is organized as follows. In Section II, the FOSDA architecture and the system operation are described. Section III reports two node-centric architectures as comparison benchmarks, and traffic statistics of two benchmark networks are analyzed to generate workloads in the comparison. The performance of FOSDA are investigated and compared with two node-centric architectures in Section IV. Section V concludes the paper by summing up the most important results.

II. FOSDA HPC NETWORK AND OPERATION

The proposed FOSDA HPC network based on distributed nanoseconds FOS and optical flow control is depicted in Figure 1. It consists of *N* racks and each rack contains *N* CPU nodes (CN), *N* memory nodes (MN), and *M* storage nodes (SN). These hardware resources are disaggregated into specific resource pools. The FOS for intra-rack communication (RFOS) connects all CPU nodes and memory nodes in the same rack to achieve high bandwidth and low latency CPU-memory communication. About latency insensitive storage traffic, an electrical packet switch (EPS) is utilized to interconnect the memory nodes and storage nodes. CPU nodes *i* (CN_i) across different racks are interconnected by the $CFOS_i$ and, similarly, memory nodes *i* (MN_i) across different racks are interconnected by the $MFOS_i$ (i = 1, …, N). As illustrated in Figure 2, an address processor and a flow controller are integrated with the hardware node in the FOSDA network. The address processor receives the resource allocation table from resource manager of FOSDA, and forwards the traffic among

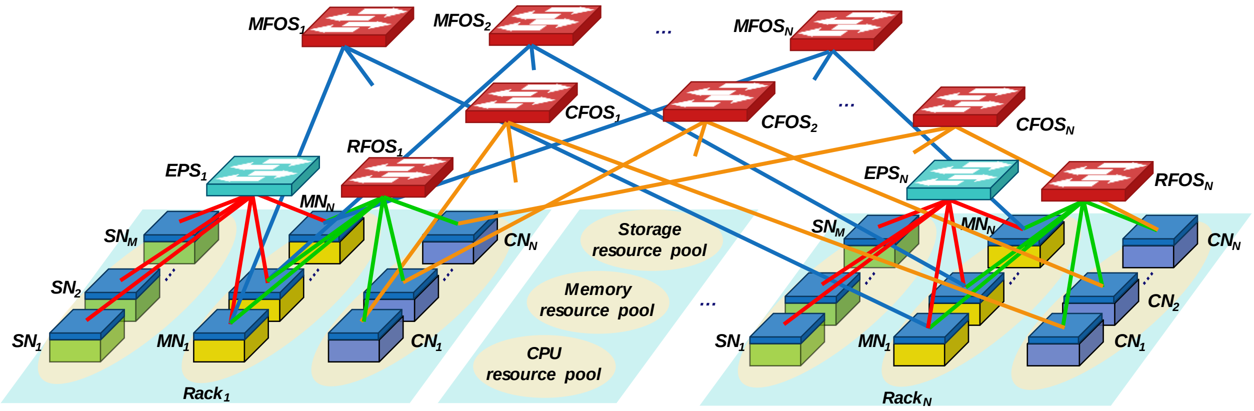


Figure 1. The FOSDA architecture.

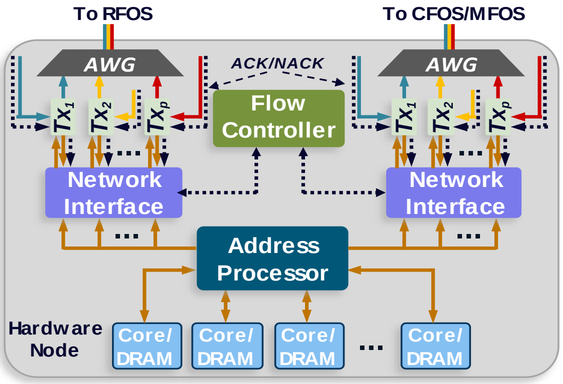


Figure 2. Fuctional blocks of the hardware node.

different resource nodes. Meanwhile, there is a flow controller in the hardware node to solve the packet contention of buffer-less optical switch [17]. The flow controller processes acknowledgement (ACK) or non-acknowledgement (NACK) signals from the FOS. The ACK signal represents the successful transmission, whereas the NACK signal means failure transmission due to packet contention. If receiving NACK signal, the flow controller sends the instruction to retransmit the packet. Both the intra-rack and inter-rack interfaces consist of *p* Wavelength-Division Multiplexing (WDM) transceivers. The *N* CPU (memory) nodes in each rack are divided into *p* groups, and each group has *F* nodes. Combining 1x*F* switch in the FOS, every TX in the hardware node serves for *F* potential destination hardware instead of *N*. Note that CPU node also contains a small memory to keep the operating system running. For the traffic between CPU nodes and memory nodes, according to the resource allocation table, the address processor maps the virtual memory address to the physical memory node address and inserts the destination address in the packet head of the CPU instruction. Then the optical packet is forwarded to RFOS/CFOS according to the address of the destination memory node. For the communication between the CPU node and the storage node, the instruction sent by the CPU node is processed by the memory controller in the memory node at first. Based on the instruction, the memory node starts reading/ writing data from/to the storage node via EPS. After finishing, the memory node sends the reply packet to the CPU node. For the traffic between memory nodes (e.g., live migration), CPU node distributes instructions to involved memory nodes first, and then memory nodes can keep on communicating without processing of CPU node.

The communication between CPU nodes and memory nodes in the same rack only crosses a single hop through the RFOS. It is also one hop for the communication between CPU/memory nodes at different racks if they are connected by the same CFOS/MFOS, while at most three hops are required to connect CPU nodes and memory nodes in the different racks. The number of hops is determined by the location of the requested data. The resource manager in the FOSDA system allocates the available memory nodes to cache the data. Based on the data request from the CPU nodes, the resource manager sets the path for the request to destination memory node. In each rack, the RFOS can support *N* CPU/memory nodes based on the *N* parallel processing modules. Meanwhile, the CFOS/MFOS interconnects the N CPU/memory nodes in N different racks. In total, *N* RFOSs are required for intra-rack communication, whereas *N* CFOS and *N* MFOS are required for inter-rack communication in the FOSDA system of *N* racks. Based on 3*N* FOS and *N* EPS, hardware resources of up to *N*×(2*N*+*M*) can be interconnected, and thus the scalability of FOSDA is guaranteed exploiting the FOS of moderate port counts.

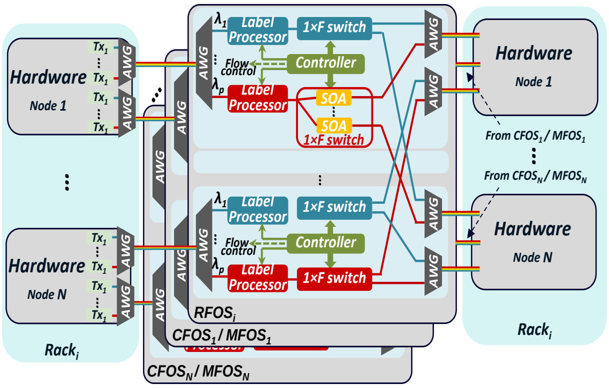


Figure 3. Schematic of the FOS.

The architecture of the FOS is illustrated in Figure 3. FOS processes the multiple WDM input packets from different resource nodes in parallel. The packets coming from the CPU/memory node i are processed by the label processor in the module *i*, where *i* = 1, ···, *N*. The optical label contains the packet destination and it is processed by the FPGA based optical switch controller. Meanwhile, the packet payload is broadcasted to the possible outputs via the 1x*F* SOA based nanoseconds optical gates. According to the lookup table distributed by the resource manager system and the optical label entry, the switch controller sets on/off the optical gates to forward or block the payload to the output destinations. If a contention happens, the packet with the highest priority is forwarded to the output, while the other packets are blocked. Then the switch controller sends the ACK/NACK signals back to the corresponding CN and MN nodes. The prototype of FOS was implemented in [18] including SOA based switch and FPGA based switch controller, which shows the feasibility of FOS. Due to the modularity and distributed processing, the FOS reconfiguration time is port count independent, allowing for low latency operation even for large scale interconnect networks.

III. SIMULATION SETUP

We investigate the workload acceptance rate, resource utilization, required hardware amount, and power consumption of FOSDA under different node-centric architectures and network scales. The statistic number of workload requests per hour is based on the Poisson distribution. Workload acceptance rate represents the rate of accepted workload requests in all the workloads requests each minute, while the request rate is defined to represent the average number of workload requests per hour. Two node-

TABLE I. POWER AND COST OF COMPONENTS IN HPC ARCHITECTURE.

Components	Specifications		
	Type	*Power (W)*	*Cost ($)*
AMD Athlon MP2000+ processor	Idle Max	115 161	149
Intel Xeon E5-2660	Idle Max	116.4 194	1329
Memory	1G 32G 96G	0.373 11.85 35.55	6.5 209 637
NIC	Wulfkit3 10Gb/s 40Gb/s 56Gb/s	14 7 10.6 11.2	180 102 338 415
Transceiver	10Gb/s 40Gb/s 56Gb/s	1 3.5 4	18 59 84
Disk	HDD	6	154
Mellanox SX6536 Switch	648ports	9073	62,125
EPS	---	2/port	20/port
FOS	12ports 18ports 48ports	77 126 489	1140 2509 17612

centric HPC architectures of different network scales are considered as benchmarks in the comparison: HPC2N [19] and iDataPlex [20]. Workloads in the comparison is realistic traces from two benchmark HPC networks. Based on workload traces from two node-centric HPC architectures, the request rates are set to 17.44 for HPC2N and 26.46 for iDataPlex respectively. The simulation is based on CloudSimPy framework [21], and hardware is a workstation of 2 Intel Xeon Gold 5118 12 cores processors, 128GB memory, and NVIDIA Quadro 16GB P5000. The operation duration of 2880 minutes is set in the comparison. The fiber distance between hardware nodes and FOS is set to 20m. Based on the FOS reconfiguration time of 43.4ns, a Round-Trip Time (RTT) of FOSDA is 243.4ns that represents the minimum latency a packet may experience. The packet processing time in the hardware node is taken as 80ns [22], including the address processing and network protocol encapsulation. The bandwidth per transceiver is set to 40Gb/s in the FOSDA. The power consumption of each architecture is calculated by the sum of component power based on an additional model [23]. The power and cost of components in diverse HPC architectures are reported in Table I, according to academic studies [24]-[27] and current commercial products [28]-[30]. Based on analyses in [26] [27], the cost of 12-port FOS is $1140, while $17612 for 48-port FOS. Note that the cost of FOS can be further reduced by integrating the FOS processing modules into Application Specific Integrated Circuit (ASIC) chip.

The performance of the FOSDA is compared with two node-centric architectures in Section IV.A. The HPC2N consisting of 120 nodes (240 cores and 120GB memory in total). Based on a WulfKit 3 Scalable Coherent Interface (SCI) network, the computing nodes in the HPC2N are connected as a three-dimensional torus switching topology of 4×5×6 grid. The network consisting three SCI rings via Peripheral Component Interconnect (PCI) bus, providing 5.4Gb/s bandwidth per port. In this comparison, the FOSDA consists of 12 racks, supporting up to 144 nodes. To keep the same amount of hardware as the HPC2N, each rack has 10 CPU/memory nodes. The splitting ratio *F* equals 4, and transceiver amount for CFOS/MFOS is 3. The iDataPlex has a larger network scale than HPC2N, including 320 nodes (2560 cores and 10240 GB memory in total). The network interconnection of the iDataPlex cluster is based on the high-performance FDR InfiniBand network, in which a Mellanox SX6536 FDR 648-ports InfiniBand Director Switch is deployed. The FOSDA is set as 18 racks for the comparison with the iDataPlex. With the splitting ratio of 6 and transceiver amount of 3 in FOSDA, there are 2 racks consisting of 16 CPU/memory nodes to keep the hardware amount the same as iDataPlex. In Section IV.B, the scalability of FOSDA is investigated under the scale of 48 racks and 2304 nodes. The transceiver amount *p* and splitting ratio *F* of the optical switch equal 6 and 8 respectively. The node-centric architecture in this case is scaling iDataPlex cluster out to 2304 nodes. The network interconnection is based on Leaf Spine network topology, consisting of 48 96-ports switches and 8 128-ports switches. Finally, in Section IV.C, the capital and operational costs of FOSDA are analyzed and compared with two node-centric architectures under specifications shown in Table I.

The Cumulative Distribution Function (CDF) of workload statistic applied in assessments is shown in Figure 4. The workload traces are from two benchmark node-centric networks. For various hardware resources in HPC networks, storage and network resources are usually sufficient to serve workload requirement, and the bottleneck is often from the performance of CPU and memory. Therefore, the traffic statistics in assessments consist of the required amount of core and memory size as well as running time. The statistic

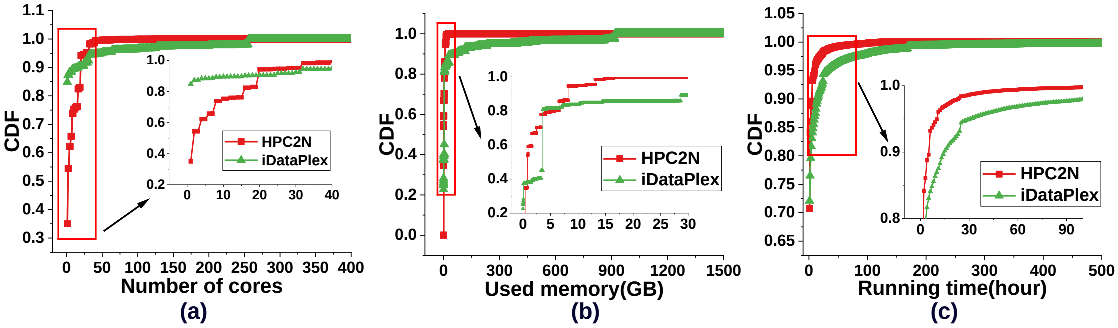


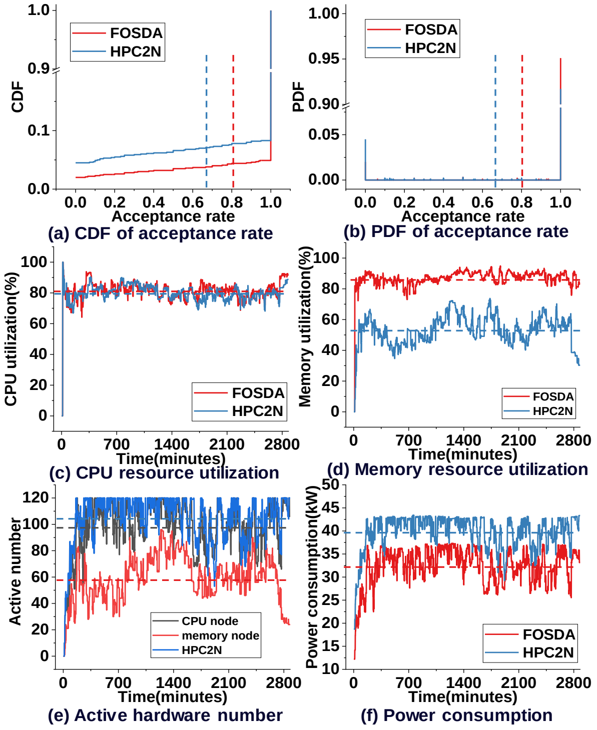
Figure 4. CDF of (a) CPU demand, (b) memory demand, and (c) running time.

of CPU demand in Figure 4(a) show that over 90% workloads have a CPU requirement of less than 50 cores in the both two architectures. Although more than 80% workloads require the CPU resource of less than 4 cores, iDataPlex also has more workloads requiring more than 250 cores CPU resource. Meanwhile, workloads have a more diverse demand of CPU resource in the HPC2N. Based on the CDF of memory demand illustrated in Figure 4(b), the memory demand in HPC2N mainly ranges from 0 and 17GB. Due to a larger network scale, iDataPlex also has 8.5% workloads with a memory demand of more than 100GB. It is shown in Figure 4(c) that more workloads require longer running time in the iDataPlex, while more than 60% workloads have a running time of less than 2 hours in two HPC networks.

IV. Assessment Results and Discussion

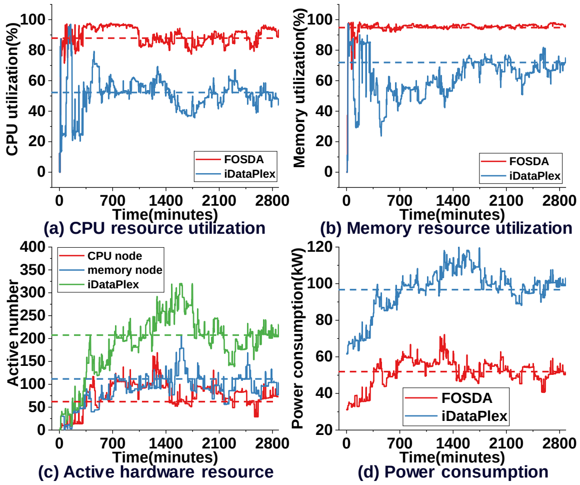
A. Comparison with Two Node-centric Architectures

Figures 5 and 6 show the workload acceptance rate, resource utilization, active hardware number, and power consumption of the FOSDA compared with two node-centric architectures under the realistic request rate. The comparison results between FOSDA and HPC2N are shown in Figure 5. As depicted in Figure 5(a) and (b), some workload requests are blocked in both the FOSDA and HPC2N due to the limited hardware resources. Despite this, the FOSDA can accept the workload requests with 80.7%, while the acceptance rate of HPC2N is 67.7%, which indicates that with the same amount of hardware, the FOSDA accepts 13% more workload requests. The reason is that, based on the independent resource allocation and fast network interconnection, the FOSDA can minimize the idle resource wasted, and has more available resources to deploy more workload request. Meanwhile, the average CPU resource utilization is 81.1% and 79.7% for FOSDA and HPC2N, respectively, as shown in Figure 5(c). The average memory resource utilization is 86.5% in FOSDA, which is also 33.4% higher compared with HPC2N in Figure 5(d). This is because the FOSDA provides more flexible resource to serve workloads with different requirements and maximize the utilization of each CPU/memory node. With much higher resource utilization and more powerful performance exploiting nanoseconds FOS, the FOSDA requires less hardware to deploy workload requests. Summarizing the active hardware resource in Figure 5(e), FOSDA only needs 97.6 CPU nodes and 58.4 memory nodes on average while HPC2N requires 103.9 computing nodes. Moreover, idle resource nodes in FOSDA are transferred into sleep mode that consume less power. As shown in Figure 5(f), the HPC2N consumes the power of 39.6kW and the FOSDA uses 18.7% less power than HPC2N.



(a) CDF of acceptance rate
(b) PDF of acceptance rate
(c) CPU resource utilization
(d) Memory resource utilization
(e) Active hardware number
(f) Power consumption

Figure 5. Comparison results between FOSDA and HPC2N.



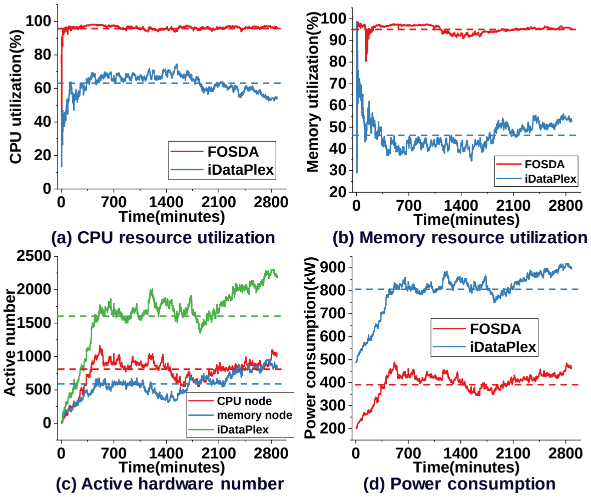
(a) CPU resource utilization
(b) Memory resource utilization
(c) Active hardware resource
(d) Power consumption

Figure 6. Comparison results between FOSDA and iDataPlex.

The iDataPlex has a larger network scale and more hardware resources than the HPC2N. Thus, under the realistic request rate of 26.46, all the workload requests are accepted in FOSDA and iDataPlex. In the comparison of the CPU utilization, FOSDA achieves 88.7% utilization, as shown in Figure 6(a), which is 36.6% higher than the iDataPlex one. Meanwhile, Figure 6(b) shows that the FOSDA also obtain 21.5% higher memory resource utilization (FOSDA 93.4% while iDataPlex 71.9%). Benefiting from fully exploiting available resources in each resource node, workloads with diverse resource requirements achieve a better resource utilization in the FOSDA than the specific hardware (CPU or memory) intensive workloads. Figure 6(c) shows that FOSDA also needs less active hardware resources than iDataPlex. FOSDA uses 62.7 CPU nodes and 112.3 memory nodes while iDataPlex requires 206.2 computing nodes on average. In terms of average power consumption, FOSDA consumes 51.1kW while iDataPlex consumes 96.1kW, which achieves 46.8% power saving, as shown in Figure 6(d). Under workload requests with diverse requirements, FOSDA minimizes the waste of the active hardware, and achieves the largest power saving.

B. Scalability of FOSDA Architecture

To investigate the scalability of the FOSDA architecture, we consider a network scale of 2304 nodes in this section, and the request rate in this case is set to 200. With more available hardware, all the workload requests are accepted in both FOSDA and node-centric architectures. The comparison results of resource utilization, active resource number, and power consumption are reported in Figure 7. The resource utilization of CPU and memory are shown in Figure 7 (a) and (b). The CPU resource utilization for FOSDA is 96.1%, which is 33.6% higher than the iDataPlex. Meanwhile, the memory resource utilization for FOSDA is 95.1%, while for iDataPlex is 46.6%. Considering the comparison of the resource utilization in several cases, it is shown that the node-centric architecture cannot achieve high CPU and memory utilization simultaneously. This is because the CPU and memory are closely coupled in the node-centric architecture. When CPU (memory) achieves high utilization (less available), the available resource of memory (CPU) in the same computing node of node-centric architecture is wasted (low utilization). In the contrast, the FOSDA can achieve both high CPU and memory utilization based on the independent resource allocation. In response to the increased workload requests, there are more running hardware resources, as illustrated in Figure 7(c). The iDataPlex has 1618 computing nodes running in average, whereas the FOSDA only requires 769 CPU nodes and 574 memory nodes in total. Moreover, as presented in Figure 7(d), the power consumption of FOSDA is 397.6kW with 2304 nodes, which is 50.4% less compared with the iDataPlex.



(a) CPU resource utilization

(b) Memory resource utilization

(c) Active hardware number

(d) Power consumption

Figure 7. Comparison results under network scale of 2304 nodes.

It is demonstrated in numerical assessments that the FOSDA architecture outperforms current node-centric HPC networks regardless of workloads and network scales. This is because that the FOSDA architecture provides flexible resource provision and avoid the bottleneck of low speed peripheral bus in node-centric networks. Exploiting the flat and fast network interconnection based on FOS, the overhead of access memory node is minimized.

C. Capital and Operational Cost Comparison

Besides the operational performance, the capital and operational cost of FOSDA are also investigated and compared with the node-centric HPC architectures. Considering the small probability of hardware fault and diverse options of the hardware upgrade, we assume that all the hardware work in the normal state in the evaluation. In this section, the operation cost of FOSDA is compared with the node-centric architectures based on the realistic workload request rate. According to the industrial electricity price reported by the European Commission [31], the average power price is $0.11/kWh. Based on the subcomponents cost in Table I and power consumption results in numerical assessments, the capital and operational costs of FOSDA and node-centric architectures are shown in Table II, in which the operation cost is calculated for one year.

TABLE II. CAPITAL AND OPERATIONAL COST OF FOSDA AND NODE-CENTRIC ARCHITECTURES.

Architectures		**Cost**	
		Capital cost (k$)	***Operation Cost/year (k$)***
FOSDA	up to 144nodes up to 324nodes	346.8 1388.3	30.6 48.7
HPC2N	120 nodes	223.4	37.6
iDataPlex	320 nodes	1114	91.3

The maximum hosted node of the FOSDA is different to the node amount of node-centric architectures under a network scale. For the fairness of comparison, the FOSDA keeps the same hardware amount as node-centric architectures in the comparison. It is shown that 12 racks scale FOSDA (up to 144 nodes) saves 18.6% operational cost per year compared with HPC2N with 120 nodes, while requiring 35.6% higher capital cost (FOSDA 346.8k$ and HPC2N 223.4k$). Meanwhile, 18 racks scale FOSDA of (up to 324 nodes) requires 46.7% less operational cost than iDataPlex, at 19.8% higher capital cost (274.3k$). Those results indicate that, as the HPC network scales, the operational cost of FOSDA increases much slower than the one of the node-centric architectures. Balancing the capital and operation cost of the HPC system, it is demonstrated that FOSDA outperforms the current node-centric architectures, especially for large scale HPC system.

V. CONCLUSIOIN

In this work, we presented a novel disaggregated HPC architecture FOSDA based on distributed nanoseconds optical switches that connects the disaggregated resources by a flat scalable optical interconnect network of high bandwidth and low latency. Numerical assessments show that, compared with node-centric HPC architectures, FOSDA can accept up to 13% more workload requests, while achieving up to 36.6% higher CPU and 21.5% higher memory utilizations with 45.5% less active hardware resources. In addition, FOSDA saves 46.8% power consumption compared with node-centric HPC architecture of 320 computing nodes. With the increment of workload requests and network scale, FOSDA presents more advantages than node-centric architecture. FOSDA obtains 33.6% higher CPU and 48.5% higher memory utilization while saving 50.4% power consumption under a network scale of 2304 nodes. Moreover, compared with the node-centric HPC architectures, FOSDA requires 46.7% less operational cost with only 19.8% higher capital cost.

REFERENCES

[1] F. Karinou, I. Roudas, K. Vlachos, B. Hemenway, and R. Grzybowski, “Influence of transmission impairments on the OSMOSIS HPC optical interconnect architecture,” J. Lightw. Technol., vol. 29, no. 21, pp. 3167–3177, 2011.

[2] Summit, https://www.olcf.ornl.gov/olcf-resources/compute-systems/summit, Retrieved: November, 2018

[3] R. Cheveresan, M. Ramsay, C. Feucht, and I. Sharapov, “Characteristics of workloads used in high performance and technical computing,” Proceedings of the 21st annual international conference on Supercomputing, New York, USA, pp. 73–82, 2007.

[4] R. Lin, Y. Cheng, M. D. Andrade, L. Wosinska, and J. Chen, ‘‘Disaggregated data centers: Challenges and trade-offs,’’ IEEE Commun. Mag., vol. 58, no. 2, pp. 20–26, Feb. 2020.

[5] O. C. Project. The Open Compute server architecture specifications [Online], http://www.opencompute.org/. Retrieved: November, 2011

[6] P. Grun, “Introduction to infiniband for end users,” [Online]. Retrieved: June, 2017, from http://www.mellanox.com/pdf/whitepapers/Intro_to_IB_for_End_Users.pdf. Accessed on: Jun. 10, 2017.

[7] Intel, “New photonic architecture promises to dramatically change next decade of disaggregated rack scale server designs,” [Online], https://newsroom.intel.com/news-releases/intel-facebook-collaborate-on-future-data-center rack-technologies. Retrieved: December, 2016.

[8] K. Lim, et al. “System-level implications of disaggregated memory,” in Proc. IEEE Int. Symp. High-Perform. Comput. Archit., Washington, USA, pp. 1–12, 2012.

[9] J. Gu, Y. Lee, Y. Zhang, M. Chowdhury, and K. Shin, “Efficient memory disaggregation with infiniswap,” in Proc. 14th USENIX Symp. Netw. Syst. Design Implement. (NSDI), Carlsbad, USA, pp. 649–667, 2017.

[10] Y. Shan, Y. Huang, Y. Chen, and Y. Zhang, “Legoos: A disseminated, distributed OS for hardware resource disaggregation,” in 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18). Carlsbad, CA, pp. 69–87, 2018.

[11] J. Kim, W. Dally, S. Scott, and D. Abts, “Technology-driven, highlyscalable dragonfly topology,” SIGARCH Comput. Archit. News, vol. 36, pp. 77–88, 2008.

[12] Y. Yan, et al. “All-optical programmable disaggregated data center network realized by FPGA-based switch and interface card,” J. Lightwave Technol., vol. 34, no. 8, pp. 1925–1932, 2016.

[13] G. Zervas, H. Yuan, A. Saljoghei, Q. Chen, and V. Mishra, “Optically disaggregated data centers with minimal remote memory latency: technologies, architectures, and resource allocation [Invited],” J. Opt. Commun. Netw., vol. 10, no. 2, pp. A270–A285, 2018.

[14] M. Bielski et al., "dReDBox: Materializing a full-stack rack-scale system prototype of a next-generation disaggregated datacenter," Design, Automation & Test in Europe Conference & Exhibition (DATE), Germany, pp. 1093-1098, 2018.

[15] P. Gao, et al. “Network requirements for resource disaggregation,” in 12th USENIX Symp. on Operating Systems Design and Implementation (OSDI), Savannah, Georgia, pp. 249–264, 2016.

[16] N. Terzenidis, M. Moralis-Pegios, G. Mourgias-Alexandris, T. Alexoudi, K. Vyrsokinos and N. Pleros, "High-Port and Low-Latency Optical Switches for Disaggregated Data Centers: The Hipoλaos Switch Architecture," Journal of Optical Communications and Networking, vol. 10, no. 7, pp. 102-116, 2018.

[17] W. Miao and N. Calabretta, “Low latency and efficient optical flow control for intra data center networks,” Opt. Express vol. 22, no. 1, pp. 427–434, 2014.

[18] X. Xue et al., "ROTOS: A Reconfigurable and Cost-Effective Architecture for High-Performance Optical Data Center Networks," Journal of Lightwave Technology, vol. 38, no. 13, pp. 3485-3494, 2020.

[19] HPC2N, https://www.hpc2n.umu.se/resources/hardware/seth. Retrieved: June, 2017.

[20] iDataPlex, https://www.pik-potsdam.de/services/it/hpc. Retrieved: January, 2019.

[21] CloudSimPy, https://github.com/RobertLexis/CloudSimPy. Retrieved: December, 2020.

[22] F. Yan, W. Miao, O. Raz and N. Calabretta, “OPSquare: A flat DCN architecture based on flow-controlled optical packet switches,” Journal of Optical Communications and Networking, vol. 9, no. 4, pp. 291-303, 2017.

[23] M. Dayarathna, Y. Wen, and R. Fan, “Data center energy consumption modeling: a survey.” IEEE Commun Surv Tutorials, vol. 18, no. 1, pp. 732–94, 2016.

[24] B. Giridhar, et al., “Exploring DRAM organizations for energy-efficient and resilient exascale memories,” in Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal. SC, pp. 23:1–23:12, 2013.

[25] J. Aroca, A. Chatzipapas, A. Anta, and V. Mancuso, “A Measurement-based Analysis of the Energy Consumption of Data Center Servers.” Proceedings of the 5th international conference on Future energy systems, UK, pp. 63-74, 2014.

[26] F. Yan, W. Miao, H. Dorren, and N. Calabretta, "On the cost, latency, and bandwidth of LIGHTNESS data center network architecture." International Conference on Photonics in Switching, Italy, pp. 130-132, 2015.

[27] X. Guo et al., "RDON: a rack-scale disaggregated data center network based on a distributed fast optical switch," Journal of Optical Communications and Networking, vol. 12, no. 8, pp. 251-263, 2020.

[28] AMD Athlon Dual-Core Processor [Online], https://www.amd.com/system/files/TechDocs/33425.pdf. Retrieved: January, 2007.

[29] Intel Xeon E5 Processor [Online], https://ark.intel.com/content/www/us/en/ark.html#@Processors. Retrieved: March, 2012.

[30] Mellanox SX6536 Switch, [Online], http://www.mellanox.com/related-docs/user_manuals/SX6536_user_manual.pdf. Retrieved: January, 2017.

[31] Energy prices and costs in Europe [Online], https://ec.europa.eu/energy/sites/ener/files/report_on_energy_prices_and_costs_in_europe_com_2020_951.pdf. Retrieved: October, 2020.

Computing Efficiency in Membrane Systems

Claudio Zandron
DISCo - Università degli Studi di Milano-Bicocca
Viale Sarca 336/14, 20126 Milano, Italy
Email: claudio.zandron@unimib.it

***Abstract*—Membrane systems (or P systems) are a computational model inspired by the functioning of the cell, and based upon the notion of cellular membrane. In this paper, we give a survey of some main results concerning the model, to show its potential to approach various problems in the area of the theory of computation.**

Keywords-Natural Computing; Membrane systems; computational complexity.

I. Introduction

Membrane systems (also known as *P systems*) have been proposed by Gh. Paun in [14] as a parallel, nondeterministic, synchronous and distributed model of computation inspired by the structure and functioning of living cells. The model consists of a hierarchical structure composed by several membranes, embedded into a main membrane called the *skin*. Membranes defines *regions* that contain multisets of *objects* (represented by symbols of an alphabet) and *evolution rules*.

Using these rules, the objects evolve and are moved from a region to a neighboring one. The rules are applied using a maximally parallel and nondeterministic semantic: all objects which can evolve in a computation step must evolve; if different sets of rules can be applied in a computation step (in a maximal parallel way), then one of them is nondeterministically chosen.

A *computation* starts from an initial configuration of the system and terminates when no evolution rule can be applied. The result of a computation is the multiset of objects contained into an *output membrane*, or emitted from the skin membrane.

The model was investigated both under theoretical aspects as well as for applications to other disciplines. In particular, the formalism is suitable to model various biological systems, thanks to its features. For instance, it allows identification of separated compartmentalized spaces where different reactions can take place; it is characterized by an easy understandable writing of reactions; it can be easily simulated in a distributed and parallel computing architecture, by separating the computation carried on by each single compartment, and simply syncronizing the exchange of information at specific time-steps. Other types of application were also investigates to different disciplines, such as cryptography, approximate optimization, or even economy. For a systematic introduction to P systems, we refer the reader to [16] [17]; recent information can be found in the dedicated webpage [32].

P systems with active membranes is a variant of the basic model introduced in [15]: in this variant, membranes can be multiplied by dividing existing ones, and the objects are communicated according to electrical charges associated with the membranes. Such features allow the construction of an exponential workspace in linear time, which can then be used in parallel to attack computationally hard problems.

In this paper we give a survey of some main results concerning theoretical aspects of the model, to show its potential in approaching various problems in the area of the theory of computation. In Section 2, we recall formal definitions related to P systems with active membranes. In Section 3, we give a very brief summary of results concerning their computating power, while in Section 4 we recall some results related to computing efficiency. Section 5 is devoted to recall some results concerning space complexity for membrane systems. Finally Section 6 draws some conclusions and suggest a few topics for research investigation.

II. Definitions

In this section, we recall the basic definition of P systems with active membranes.

Definition 1: A *P system with active membranes* of initial degree $d \geq 1$ is a tuple $\Pi = (\Gamma, \Lambda, \mu, w_1, \ldots, w_d, R)$, where:

- Γ is an alphabet, a finite non-empty set of symbols, usually called *objects*;
- Λ is a finite set of labels for the membranes;
- μ is a membrane structure (i.e., a rooted *unordered* tree) consisting of d membranes enumerated by $1, \ldots, d$; each membrane is labeled by an element of Λ, not necessarily in a one-to-one way, and possesses an *electrical charge* (or polarization), that can be neutral (0), positive (+) or negative (−).
- $w_1, \ldots, w_d$ are strings over Γ, describing the initial multisets of objects placed in the d regions of μ;
- R is a finite set of rules.

The rules are of the following kinds:

- *Object evolution rules*, of the form $[a \to w]_h^\alpha$
  They can be applied if the membrane h has charge α and contains an occurrence of the object a; the object a is rewritten into the multiset w.
- *Send-in communication rules*, of the form $a\,[\;]_h^\alpha \to [b]_h^\beta$
  They can be applied to a membrane labeled by h, having charge α and if the external region contains an occurrence of the object a; the object a is sent into h becoming b and, simultaneously, the charge of h is changed to β.
- *Send-out communication rules*, of the form $[a]_h^\alpha \to [\;]_h^\beta\, b$
  They can be applied to a membrane labeled by h, having charge α and containing an occurrence of a; the object a is sent out from h to the outside region becoming b. Simultaneously, the charge of h is changed to β.

ISBN: 978-1-61208-846-4

- *Dissolution rules*, of the form $[a]^\alpha_h \to b$
  They can be applied to a membrane labeled by h, having charge α and containing an occurrence of the object a; the membrane h is dissolved and its contents are left in the surrounding region unaltered, except that an occurrence of a becomes b.
- *Elementary division rules*, of the form $[a]^\alpha_h \to [b]^\beta_h\,[c]^\gamma_h$
  They can be applied to a membrane labeled by h, having charge α, containing an occurrence of the object a but having no other membrane inside (an *elementary membrane*); the membrane is divided into two membranes having label h and charge β and γ; the object a is replaced, respectively, by b and c while the other objects in the initial multiset are copied to both membranes.
- *Nonelementary division rules*, of the form

$$\big[[\,]^+_{h_1}\cdots[\,]^+_{h_k}[\,]^-_{h_{k+1}}\cdots[\,]^-_{h_n}\big]^\alpha_h \to$$

$$\big[[\,]^\delta_{h_1}\cdots[\,]^\delta_{h_k}\big]^\beta_h\big[[\,]^\epsilon_{h_{k+1}}\cdots[\,]^\epsilon_{h_n}\big]^\gamma_h$$

  They can be applied to a membrane labeled by h, having charge α, containing the positively charged membranes $h_1, \ldots, h_k$, the negatively charged membranes $h_{k+1}, \ldots, h_n$, and possibly some neutral membranes. The membrane h is divided into two copies having charge β and γ, respectively; the positively charged membranes $h_1, \ldots, h_k$ are placed inside the former membrane, their charge set to δ, while the negative ones are placed inside the latter membrane, their charges set to ϵ. Neutral membranes inside h are duplicated and placed inside both copies.

Each instantaneous configuration of a P system with active membranes is described by the current membrane structure, including the electrical charges, together with the multisets located in the corresponding regions. A computation step changes the current configuration according to the following set of principles:

- Each object and membrane can be subject to at most one rule per step, except for object evolution rules (inside each membrane any number of evolution rules can be applied simultaneously).
- The application of rules is *maximally parallel*: each object appearing on the left-hand side of evolution, communication, dissolution or elementary division rules must be subject to exactly one of them (unless the current charge of the membrane prohibits it). The same reasoning applies to each membrane that can be involved to communication, dissolution, elementary or nonelementary division rules. In other words, all possible rules that can be applied must be applied at each computation step; the only objects and membranes that do not evolve are those associated with no rule, or only to rules that are not applicable due to the electrical charges.
- When several conflicting rules can be applied at the same time, a nondeterministic choice is performed; this implies that, in general, multiple possible configurations can be reached after a computation step (e.g., consider two rules $a \to b$ and $a \to c$ in a region h; if an object a is present in that region, then it can nondeterministically produce either b or c, by using respectively the first or the second rule).
- While all the chosen rules are considered to be applied simultaneously during each computation step, they are logically applied in a bottom-up fashion: first, all evolution rules are applied to the elementary membranes, then all communication, dissolution and division rules; then the application proceeds towards the root of the membrane structure. In other words, each membrane evolves only after its internal configuration has been updated.
- The outermost membrane cannot be divided or dissolved, and any object sent out from it cannot re-enter the system again.

A *halting computation* of the P system Π is a finite sequence of configurations $\vec{\mathcal{C}} = (\mathcal{C}_0, \ldots, \mathcal{C}_k)$, where $\mathcal{C}_0$ is the initial configuration, every $\mathcal{C}_{i+1}$ is reachable by $\mathcal{C}_i$ via a single computation step, and no rules can be applied anymore in $\mathcal{C}_k$. The result of a halting computation is the multiset of objects emitted from the skin during the whole computation. A *non-halting* computation $\vec{\mathcal{C}} = (\mathcal{C}_i : i \in \mathbb{N})$ consists of infinitely many configurations, again starting from the initial one and generated by successive computation steps, where the applicable rules are never exhausted. A non–halting computation produces no output.

P systems can also be used as *recognizers* (see, e.g., [3]) by employing two distinguished objects `yes` and `no`; exactly one of these must be sent out from the outermost membrane during each computation, in order to signal acceptance or rejection respectively; we also assume that all computations are halting. If all computations starting from the same initial configuration are accepting, or all are rejecting, the P system is said to be *confluent*. If this is not necessarily the case, then we have a *non–confluent* P system, and the overall result is established as for nondeterministic Turing machines: it is acceptance iff an accepting computation exists. All P systems considered in this paper are confluent.

In order to solve decision problems (i.e., decide languages), we use *families* of recognizer P systems $\mathbf{\Pi} = \{\Pi_x : x \in \Sigma^\star\}$. Each input x is associated with a P system Π_x that decides the membership of x in the language $L \subseteq \Sigma^\star$ by accepting or rejecting. The mapping $x \mapsto \Pi_x$ must be efficiently computable for each input length [13].

Definition 2: A family of P systems $\mathbf{\Pi} = \{\Pi_x : x \in \Sigma^\star\}$ is said to be *(polynomial-time) uniform* if the mapping $x \mapsto \Pi_x$ can be computed by two deterministic polynomial-time Turing machines F (for “family”) and E (for “encoding”) as follows:

- The machine F, taking as input *the length* n *of* x in unary notation, constructs a P system Π_n, which is common for all inputs of length n, with a distinguished input membrane.
- The machine E, on input x, outputs a multiset w_x (an encoding of x).
- Finally, Π_x is simply Π_n with w_x added to the multiset placed inside its input membrane.

Definition 3: If the mapping $x \mapsto \Pi_x$ is computed by a *single* polynomial-time Turing machine, the family $\mathbf{\Pi}$ is said to be *semi-uniform*. In this case, inputs of the same size may be associated with P systems having possibly different membrane structures and rules.

Any explicit encoding of Π_x is allowed as output of the construction, as long as the number of membranes and objects represented by it does not exceed the length of the whole description, and the rules are listed one by one. This restriction is enforced to mimic a (hypothetical) realistic process of construction of the P system, where membranes and objects are placed in a constant amount during each construction step, and require actual physical space proportional to their number. Moreover, notice that uniformity condition can also be restricted to be computed in classes below $\mathbf{P}$, such as log–space Turing machines. We refer the reader to [13] for further details on the encoding of P systems.

Finally, we describe how time and space complexity for families of recognizer P systems are measured.

Definition 4: A uniform or semi–uniform family of P systems $\mathbf{\Pi} = \{\Pi_x : x \in \Sigma^\star\}$ is said to decide the language $L \subseteq \Sigma^\star$ (in symbols $L(\mathbf{\Pi}) = L$) in time $f \colon \mathbb{N} \to \mathbb{N}$ iff, for each $x \in \Sigma^\star$,

- the system Π_x accepts if $x \in L$, and rejects if $x \notin L$;
- each computation of Π_x halts within $f(|x|)$ computation steps.

Definition 5: Let $\mathcal{C}$ be a configuration of a P system Π. The *size* $|\mathcal{C}|$ *of* $\mathcal{C}$ is defined as the sum of the number of membranes in the current membrane structure and the total number of objects they contain. If $\vec{\mathcal{C}} = (\mathcal{C}_0, \ldots, \mathcal{C}_k)$ is a halting computation of Π, then the *space required by* $\vec{\mathcal{C}}$ is defined as

$$|\vec{\mathcal{C}}| = \max\{|\mathcal{C}_0|, \ldots, |\mathcal{C}_k|\}$$

or, in the case of a non-halting computation $\vec{\mathcal{C}} = (\mathcal{C}_i : i \in \mathbb{N})$,

$$|\vec{\mathcal{C}}| = \sup\{|\mathcal{C}_i| : i \in \mathbb{N}\}.$$

Non-halting computations might require an infinite amount of space (in symbols $|\vec{\mathcal{C}}| = \infty$). The *space required by* Π itself is then

$$|\Pi| = \sup\{|\vec{\mathcal{C}}| : \vec{\mathcal{C}} \text{ is a computation of } \Pi\}.$$

Notice that $|\Pi| = \infty$ occurs if either Π has a non-halting computation requiring infinite space, or Π has an infinite set of halting computations, such that for each bound $b \in \mathbb{N}$ there exists a computation requiring space larger than b.

III. Computing Power of Membrane Systems

The first studies of the model concerned its computing power: various types of Membrane systems have been compared with computing models like automata, Turing machines and register machines.

It is known that using a single membrane we can only generate the length sets of context–free languages, and the power cannot be extended by using an unlimited number of membranes. However, if we allow to dissolve membranes after the application of rewriting rules then the computing power is increased, when at least two membranes are used.

More formally, let us denote by $NOP_k(\delta)$ (resp. $NOP_k(n\delta)$) the family of natural numbers generated by P systems having k membranes and using (resp. not using) the dissolving membrane action. The following results can be stated [16]:

Theorem 1: $NOP_1(n\delta) = NOP_*(n\delta) = NCF$
$NCF = NOP_*(n\delta) \subset (NE0L \subseteq) NOP_2(\delta)$
$NOP_*(\delta)(\subseteq ET0L) \subset NCS$

Even when dissolving membrane action is allowed, universality cannot be reached. Some more ingredients can be considered to obtain such a result like, e.g., cooperative rules, catalysts, or priorities defining the order of rules application.

A rewriting rule $a \to w$ is said to be cooperative if a contains more than one symbol. This turned out to be a feature very powerful in the framework of membrane systems. In fact, when using such rules one membrane turns out to be sufficient to obtain the same power as Turing machines:

Theorem 2: $NOP_1(coop) = NOP_*(coop) = NRE$

where *coop* indicates the possibility to use cooperative rules.

A simpler form of cooperative rules can be defined by means of catalysts. A rewriting rule with catalyst is a rule of the form $CX \to Cw$, where C and X are symbols and w is a string. C is said to be a catalyst: it is needed to activate the rule, but it is not changed by it. Such a feature also allows to obtain universal systems, but only when priorities defining a partial order concerning the application of the rules is also used:

Theorem 3: $NOP_2(cat, pri) = NOP_*(cat, pri) = NRE$

One can also consider structured objects instead of atomic ones, by consider strings of symbols: in this case, the systems are called Rewriting P systems. Let us denote by RP_k the family of languages generated by Rewriting P systems using k membranes and context–free rewriting rules. The following theorem from [16] shows that using a single membrane only context–free languages can be obtained, but a structure with four membranes allow to obtain a strictly more powerful class.

Theorem 4: $RP_1(CF) = CF \subset RP_4(CF)$

Thus, it is evident from these results that the power of such systems can be improved (as expected) by exploiting membranes to define regions to keep separated specific subsets of rules and objects. Once again, universality cannot be obtained using only this basic set of ingredients, and more features must be considered.

Further details can be found in [16] and [17].

IV. Computing Efficiency of Membrane Systems

Another interesting feature that can be considered, and already described in Section 2, is the possibility to give an active role to membranes. P systems with active membranes allow to create new membranes during the computation by division of existing membranes. In this way, we can obtain a trade off between time and space resources that allows to solve NP–complete (or even harder) problems in polynomial time and exponential space (see, e.g., [15] [7] [8] [29] [31]).

Theorem 5: The SAT problem can be solved in linear time (with respect to the number of variables and the number of clauses) by a confluent P-system with active membranes using elementary membrane division only.

In fact, consider a boolean expression Φ in conjunctive normal form, with m clauses and n variables. We can build a P-system $\Pi = (\Gamma, \Lambda, \mu, w_1, w_2, R)$ having initial objects $a_1, a_2, \ldots, a_n$ in region 2 and such that R is defined to contain a polynomial number of rules (with respect to the size of the input formula) that operate as it follow.

By using the variables a_i and elementary membrane division rules, in $O(n)$ steps we generate 2^n copies of membrane 2, containing all possible truth assignments of the n variables of Φ.

Then, in $O(m)$ steps we verify if there is at least one membrane containing a truth assignment that satisfies all the m clauses of Φ. In this case, an object yes is sent out from the skin membrane; otherwise, an object no is sent out.

Let us denote by $PMC_{\mathcal{NAM}}$, $PMC_{\mathcal{EAM}}$, and $PMC_{\mathcal{AM}}$ the class of problems solved in a polynomial number of steps (with respect to the input length) by P systems with active membranes without membrane division, with division for elementary membranes only, and for both elementary and non–elementary membranes, respectively.

The following results can be obtained directly from definitions:

Theorem 6: $PMC_{\mathcal{N}AM} \subseteq PMC_{\mathcal{E}AM} \subseteq PMC_{\mathcal{A}M}$

Moreover, it is easy to show that

Theorem 7: $P \subseteq PMC_{\mathcal{N}AM}$

In fact, the "trick" is that the deterministic Turing machine deciding $L \in P$ is used to solve directly the problem in polynomial time. Then, we build a P system with a single membrane containing either an object YES, whenever an input $x \in L$ is given, or NO, otherwise. This requires polynomial time, and then the P system simply send out the object in a single computation step.

The opposite inclusion is also true:

Theorem 8: $PMC_{\mathcal{N}AM} \subseteq P$

In fact, a generic P system Π without membrane division can be simulated by a deterministic Turing machine M, with a polynomial slowdown, as proved in [29].

Since we have shown that the NP-complete problem SAT can be solved when elementary membrane division is allowed, then we can also state the following:

Theorem 9: $NP \subseteq PMC_{\mathcal{EAM}}$

From this result and from the closure properties for $PMC_{\mathcal{EAM}}$ it also follows:

Theorem 10: $coNP \subseteq PMC_{\mathcal{EAM}}$

P systems with elementary membrane division can be simulated by Deterministic Turing machines using polynomial space: $PMC_{\mathcal{E}AM} \subseteq PSPACE$. Hence:

Theorem 11: $NP \cup coNP \subseteq PMC_{\mathcal{E}AM} \subseteq PSPACE$

A stronger result was later proved [19]: a solution to the PP–complete problem SQRT–3SAT was obtained using P systems with active membranes and elementary membrane division. This proved that the class PP (Probabilistic Polynomial time: the class of decision problems solvable by a probabilistic Turing machine in polynomial time, with an error probability of less than 1/2 for all instances) is also included in $PMC_{\mathcal{EAM}}$. A characterization of the class $P^{\#P}$ was obtained in [10].

When division for non-elementary membranes is allowed, even harder problems can be solved, as expected. In a series of papers [2] [26] and [24] the following results were proved:

Theorem 12: $PSPACE \subseteq PMC_{\mathcal{AM}} \subseteq EXPTIME$

By limiting the nesting levels of membranes (and, as a consequence, the membrane division) to a constant depth, the problems in the class CH (Counting Hierarchy) can solved, as proved in [23].

V. Space complexity of P–systems and polarization of membranes

In order to clarify relations between the amount of time and space needed to solve various classes of problems, in [18] a definition of space complexity for P systems has been introduced.

On the same line of what has been done for time complexity, we can define space complexity classes for Membrane systems. By $MCSPACE_T(f)$ we denote the class of languages decided by confluent recognizer P systems (of type T) within space $f(n)$. In particular, by $PMCSPACE_T (= MCSPACE_T^{[*]}(p(n)))$ we denote the class of languages decided by confluent recogniser P systems using at most a polynomial number of elements.

From the definitions and from the results we recalled in the previous section, it is easy to see that:

- $P \subseteq MCSPACE_{\mathcal{N}AM}(O(1))$
- $NP \cup co-NP \subseteq EXPMCSPACE_{\mathcal{E}AM}$
- $PSPACE \subseteq EXPMCSPACE_{\mathcal{A}M}$

In [20] and [21] it has been shown, respectively, that the $PSPACE$–complete problem Quantified–3SAT can be solved by P–systems with active membranes using a polynomial amount of space, and that such P systems can be simulated by Turing machines with only a polynomial increase in space requirements, thus giving a precise characterization of the class $PSPACE$ in terms of space complexity classes for membrane systems. A similar result to characterize the complexity class $EXPSPACE$ can be obtained by considering exponential space P systems, as showed in [1]. Thus, all types of Membrane systems with active membranes and both divisions for elementary and non-elementary membranes, and working in a polynomial space, exactly characterize $PSPACE.$

Investigation of classes of problems solved by P systems using sublinear space was also considered. In order to consider sublinear space, two distinct alphabets were considered in the definition of P systems: an *INPUT* alphabet and a *WORK* alphabet. Objects from the INPUT alphabet cannot be rewritten and do not contribute to the size of the configuration of a P system. The size of a configuration was defined as the sum of the number of membranes in the current membrane structure and the total number of working objects they contain; the space required by a computation is the maximum size among all configurations. Moreover, we need to define a uniformity condition for the families of P systems that is weaker than the usual **P** uniformity, to avoid the possibility to solve a problem directly by using the Turing machine that build the P systems we use to compute. We consider **DLOGTIME**-uniformity, defined on the basis of **DLOGTIME** Turing machines [12]. We refer the reader to [25] for formal definitions.

The efficient simulation of logarithmic space Turing machines (or other equivalent models) by employing standard techniques used in the papers previously cited seems not to work because of two main problems: we either need to use a polynomial number of working objects (thus violating the logarithmic space condition) or to use a polynomial number of rewriting rules (thus violating the uniformity condition). Nonetheless, it has been showed in [25] that such a simulation can be efficiently done by using membrane polarization both to communicate objects through membranes as well as to store some information:

Theorem 13: Each log–space deterministic Turing machine M can be simulated by a **DLOGTIME**-uniform family $\mathbf{\Pi}$ of confluent recognizer P systems with active membranes in logarithmic space.

An immediate corollary of Theorem 13 is that the class L (the class of problems solved by log–space Turing machines) is contained in the class of problems solved by **DLOGTIME**-uniform, log–space P systems with active membranes.

VI. Conclusions

We survey some results concerning P systems with active membranes, concerning both the computational aspects, as well as computational efficiency. Some results concerning space complexity of the model have also been recalled. For further reading on the subject, we refer the reader to the main volumes on the subject [16] [17].

A recent survey on different strategies to approach computationally hard problems (containing links to various research paper on the subject) by P systems with active membranes can be found in [27].

Links to various paper concerning space complexity for membrane systems are available in [20] [21]; readers interested in results concerning sublinear space or even constant amount of space can refer to [22] and [9], respectively. A recent survey concerning results obtained by considering different bounds on space can be found in [28].

There are various research topics which deserve to be investigated with respect to computing efficiency. In particular, precise characterizations of complexity classes obtained by considering systems using specific subset of features, as well as relations of such classes with standard complexity classes, both concerning time complexity and space complexity.

More general research directions actually under development concern the application of the model to describe different biological processes. As already pointed out, since Membrane systems are a bio-inspired computing model, it is natural to apply it to the description and simulation of complex bio-processes, to gain various kind of information on such processes.

Another interesting research directions concern the link of P systems with neural computing: a variant of neural-like P systems have been introduced in [6] under the name of Spiking P systems, and its application to the field of Artificial Intelligence and deep learning strategies is currently under development.

Acknowledgements

This work was partially supported by Università degli Studi di Milano-Bicocca, Fondo di Ateneo Quota Dipartimentale (FAQD-2019).

References

[1] A. Alhazov, A. Leporati, G. Mauri, A. E. Porreca, C. Zandron, Space complexity equivalence of P systems with active membranes and Turing machines, Theoretical Computer Science 529, 2014, 69—81

[2] A. Alhazov, C. Martin-Vide, L. Pan, Solving a PSPACE–complete problem by P–systems with restricted active membranes, Fund. Inf. 58, 2, 2003, 67–77

[3] E. Csuhaj-Varju, M. Oswald, Gy. Vaszil, P automata, Handbook of Membrane Computing, Gh. Paun et al. (Eds.), Oxford University Press, 2010, 144–167

[4] M. Gutierrez-Naranjo, M.J. Perez-Jimenez, P–systems with active membranes, without polarizations and without dissolution: a characterization of P, Unconv. Comp. 2005, C.S. Calude et al. (eds.), LNCS 3699, Springer, 2005, 105-116.

[5] M. Gutierrez-Naranjo, M. J. Perez-Jimenez, A. Riscos-Nunez, F. J. Romero-Campero, Characterizing tractability by cell-like membrane systems, in K.G. Subramanian et al. (Eds.), Formal Models, Languages and Applications, Ser. Mach. Percept. Artif. Intell., vol. 66, World Scientific, 2006, 137-–154

[6] M. Ionescu, Gh. Paun, T. Yokomori, Spiking neural P systems, Fundamenta Informaticae, 71, 2-3, 2006, 279–308

[7] S. N. Krishna, R. Rama, A variant of P-systems with active membranes: Solving NP-complete problems, Rom. J. of Inf. Sci. and Tech., 2, 4 (1999)

[8] A. Leporati, C. Zandron, M. A. Gutierrez-Naranjo, P systems with input in binary form, Int. J. of Found. of Comp. Sci., 17(1), 2006, 127–146

[9] A. Leporati, L. Manzoni, G. Mauri, A. E. Porreca, C. Zandron, Constant-space P systems with active membranes, Fundamenta Informaticae 134(1–2), 2014, 111–128

[10] A. Leporati, L. Manzoni, G. Mauri, A.E. Porreca, C. Zandron, Simulating elementary active membranes with an application to the P conjecture, LNCS 8961, Springer, 2014, 284–299

[11] A. Leporati, G. Mauri, C. Zandron, Quantum Sequential P Systems with Unit Rules and Energy Assigned to Membranes, in R. Freund et al (eds.), Membrane Computing, 6th Int. Work., WMC 2005, Vienna, Austria, LNCS 3850, Springer, 2006, 310–325

[12] D.A. Mix Barrington, N. Immerman, H. Straubing, On uniformity within NC^1. Journal of Computer and System Sciences 41(3), 1990, 274–306

[13] N. Murphy, D. Woods, The computational power of membrane systems under tight uniformity conditions, Natural Computing 10(1), 2011, 613–632

[14] Gh. Păun, Computing with membranes, J. of Computer and System Sciences, 61(1), 2000, 108–143

[15] Gh. Păun, P systems with active membranes: Attacking NP-complete problems, J. of Automata, Languages and Combinatorics 6(1), 2001, 75–90

[16] Gh. Păun, Membrane Computing. An Introduction, Springer, Berlin, 2002

[17] Gh. Păun, G. Rozenberg, A. Salomaa (Eds.), Handbook of Membrane Computing, Oxford University Press, 2010

[18] A. E. Porreca, A. Leporati, G. Mauri, C. Zandron, Introducing a space complexity measure for P systems, Int. J. of Comp. Comm. and Control, 4(3), 2009, 301–310

[19] A. E. Porreca, A. Leporati, G. Mauri, C. Zandron, P systems with Elementary Active Membranes: Beyond NP and coNP, in Gheorghe M. et al. (eds.), CMC 2010, Jena, Germany, August 2010, LNCS 6501, Springer, 2011, 338–347

[20] A. E. Porreca, A. Leporati, G. Mauri, C. Zandron, P Systems with Active Membranes: Trading Time for Space, Natural Computing 10(1), 2011, 167–182

[21] A. E. Porreca, A. Leporati, G. Mauri, C. Zandron, P systems with active membranes working in polynomial space, Int. J. Found. Comp. Sc., 22(1), 2011, 65–73

[22] A. E. Porreca, A. Leporati, G. Mauri, C. Zandron, Sublinear-space P systems with active membranes. In E. Csuhaj-Varjú, M. Gheorghe, G. Rozenberg, A. Salomaa, G. Vaszil, G. (eds.) Membrane Computing, 13th International Conference, CMC 2012, LNCS 7762, 2013, 342–357

[23] A. E. Porreca, L. Manzoni, A. Leporati, G. Mauri, C. Zandron, Membrane division, oracles, and the counting hierarchy, Fundamenta Informaticae, 138, 1-2, 2015, 97–111

[24] A. E. Porreca, G. Mauri, C. Zandron, Complexity classes for membrane systems, RAIRO-Theor. Inform. and Applic. 40(2), 2006, 141–162

[25] A. E. Porreca, C. Zandron, A. Leporati, G. Mauri, Sublinear Space P systems with Active Membranes, Membrane Computing: 13th International Conference, LNCS, CMC 2012, Springer, Berlin, 2013, 342–357

[26] P. Sosik, The computational power of cell division in P systems: Beating down parallel computers?, Natural Computing, 2(3), 2003, 287–298

[27] P. Sosik, P systems attacking hard problems beyond NP: a survey. Journal of Membrane Computing 1(3), 2019, 198—208

[28] C. Zandron, Bounding the space in P systems with active membranes, Journal of Membrane Computing 2(2), 2020, 137–145

[29] C. Zandron, C. Ferretti, G. Mauri, Solving NP-complete problems using P systems with active membranes, In I. Antoniou, C.S. Calude, M.J. Dinneen, eds., Unconventional Models of Computation, Springer–Verlag, London, 2000, 289–301

[30] C. Zandron, C. Ferretti, G. Mauri, Two Normal Forms for Rewriting P systems, in Machines, Computations and Universality, Proc. of 3rd Int. Conf. MCU 2001, LNCS 2055, Springer-Verlag, 2001, 153–164

[31] C. Zandron, A. Leporati, C. Ferretti, G. Mauri, M. J. Pérez-Jiménez, On the Computational Efficiency of Polarizationless Recognizer P Systems with Strong Division and Dissolution, Fundamenta Informaticae, 87(1), 2008, 79-91

[32] The P systems Web page: http://ppage.psystems.eu/

Data Pre-processing and Clustering Algorithm for Epidemic Disease Diagnosis Data

Yaoyao Sang
College of Information Science and Engineering
University of Jinan
Jinan Shandong China
email:1034168135@qq.com

Lianjiang Zhu
College of Information Science and Engineering
University of Jinan
Jinan Shandong China
email:ise_zhulj@ujn.edu.cn

Tao Du
College of Information Science and Engineering
University of Jinan
Jinan Shandong China
email:ise_dut@ujn.edu.cn

Shouning Qu
College of Information Science and Engineering
University of Jinan
Jinan Shandong China
email:qsn@ujn.edu.cn

Abstract**—The paper mainly solves the problem of nonstandard tuberculosis data, and makes cluster analysis. It is an evaluation of existing work. The current epidemic disease data are huge and has great research value, but it is not easy to be directly used to discover knowledge. In this paper, the tuberculosis clinic data is pre-processed first; the ways of pre-processing mainly includes data cleaning, data integration, data transformation and data normalization. When processing the location information in the data, an innovative method of data weighting is proposed, which makes the complex medical data be numerical and normalized. Then, the novel unsupervised machine learning method Density peak clustering algorithm is used for clustering the data set and prove the validation of our method. This work can form clustering results and discover knowledge on this basis.**

Keywords—Tuberculosis clinic data; data normalization; location information; DPC.

I. INTRODUCTION

Nowadays, the demand for intelligent medical and health care is increasing day by day. The application of big data analysis and mining in the medical field includes many directions, such as individual and group medical planning, disease management, remote patient monitoring, etc. [1]. The dataset obtained from the patient visit records is the most objective and can reflect the real situation. The transmission pattern and trend of infectious diseases can be analyzed from it by data mining. For health care workers, big data analysis and application is conducive to the prevention and treatment of epidemics. therefore, it is of practical significance to the intelligent processing and data mining of medical data. However, most of the existing medical datasets are not standardized and incomplete, and the data types are diversified. Correspondingly, the processing results are easily disturbed and negatively affected by noise value, missing value, outliers and different types of data. For data mining, low-quality data will lead to low-quality mining results [2]. Therefore, how to conduct scientific and standardized processing of massive medical data has become a hot issue, which is also the basis and necessary condition for mining valuable medical information. In order to solve the above problems, this paper proposes a set of numerical processing system for epidemic disease treatment data, including data cleaning, data integration, maximum and minimum normalization [3], and unique heat coding and weighted numerical method. In addition, it innovatively proposes a method of setting initial values and weighting them to highlight the differences, and weight the data with strong relevance to the dimension where the sub-type data are located, so as to scientifically convert them into values in the range of 0-1.

The remaining of this paper is structured as follows. Section I mainly introduces the background and significance of the subject. In Section II, Literature related to the research content is listed. In Section III, the method and process of data preprocessing are introduced, and the innovative geographic location processing method is mainly elaborated. In Section IV, the main content is the application of density peak algorithm in standardized data set. Section V mainly summarizes the content of the paper and look forward the future work.

II. RELATED WORKS

In view of the fact that most data streams are mixed attribute data, researchers have also proposed some algorithms to directly process mixed attribute data. Zhang et al.[4] proposed a hybrid algorithm combining fuzzy clustering with particle swarm optimization (PSO) for incomplete data clustering, and missing attributes are represented as intervals, which are based on Clustream [5], They proposed the micro-clustering histogram representation for the classification property of mixed attributes, and used Poisson process to model the arrival time of samples. For the clustering problem of mixed attribute data, the existing algorithm adopted the following methods. One is to convert the type attribute to numeric attribute. In this way, the data sets can be transformed into a numeric attribute dataset, which can then be processed by clustering the numeric attribute data. Shih et al. [6] converted classification attributes into numerical attributes according to the similarity of classification attributes, and then adopted k-means algorithm for clustering. In literature [7], classification attributes were converted into numerical attributes by the transformation method of spherical coordinates, and k-means algorithm was used to cluster the results. In this way, the dataset was transformed from mixed type to sub-type, and then processed by clustering method of sub-type attribute data. Rodríguez-Jiménez et al. [8] proposed new methods to generate the lattice concept with positive and negative information to be used as a kind of map of attribute connections. David and Averbuch [9] converted numerical attributes into classification attributes through CH index, and used spectral clustering to process datasets. In order to realize clustering analysis of the typed data of set-valued attributes, Giannotti [10] proposed a Trk-means algorithm based on Jaccard distance. However, there is no further analysis on the convergence of the algorithm. In order to improve the clustering algorithm for typed data, Cao et al. [11] designed and implemented the sv-k-modes clustering algorithm; this paper proposed a comprehensive clustering analysis for the data of subtype of set-valued attribute. Wangchamhan et al. [12] used Gower distance measurement to measure mixed attribute data in order to overcome the limitation that the k-means algorithm was unable to effectively process the data of different types. Chen and He [13] proposed an adaptive peak density clustering algorithm for mixed attribute data. According to the analysis of data attribute relationship, the mixed attribute dataset was classified into three types, including digital dominant data, classified dominant data and balanced data, and the corresponding distance measurement was designed. Ding et al. [14] proposed a density peak clustering algorithm based on entropy to process mixed data with fuzzy neighborhood. A new similarity measure was proposed for the numerical type with uniform standard, the similarity of numerical part is regarded as the whole and calculate the similarity of sub-type parts separately. This shows that it is challenging to process mixed attribute data by clustering algorithm alone, and most scholars focus on algorithm innovation to process data. There are few innovative points in data pre-processing, in particular, there are few studies on the relatively large amount of medical data. Therefore, this paper aims at making innovations in data preprocessing and proposes a weighted numerical processing method for geographical location data.

I. METHOD AND REALIZATION OF MEDICAL DATA PRE-PROCESSING

In this section, we will present the Data Processing Methods and Processes.

A. Data Processing Methods and Processes

Currently, thousands of records about desensitized TB patients are given, which are respectively based on six municipal cities in a province. Statistical analysis was conducted in each municipal city based on patients, including basic personal information such as gender, age and home address, as well as disease information such as treatment time, patient source, diagnosis location and result. The overall information is relatively comprehensive and complete, which could effectively complete subsequent data pre-processing and data mining.

Figure 1 is an overall flow chart of treatment methods for epidemic disease treatment data. Considering the diverse types of treatment data and the large proportion of categorical values, appropriate data pre-processing methods are selected to convert the original data into a numerical type. This step lays a foundation for subsequent data mining.

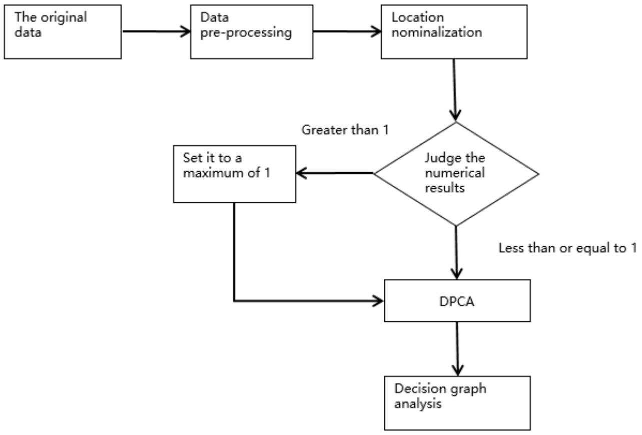


Figure. 1. Overall flow chart of treatment method for epidemic disease diagnosis data

B. Analysis of Data Pre-Processing Steps

The data pre-processing shown in Figure 1 is mainly divided into the following steps：

1) In the data pre-processing of the original medical data set, data cleaning was first carried out, which mainly to delete the irrelevant and duplicate data, and deal with missing values and outliers. According to the practical significance, if the missing value accounts for less than 2% and is not easy to fill, it can be directly deleted or ignored. However, some missing values will affect the machine processing results of the whole dataset. At this time, Lagrange interpolation method is selected for interpolation, and the formula of Lagrange interpolation is shown in (1).

$$L(x) = \sum_{i=0}^{n} y_i \prod_{j=0,j\neq i}^{n} \frac{x-x_j}{x_i-x_j} \tag{1}$$

Where, $x_i$is the x-coordinate of the known points, $x_j$is the function value corresponding to the known points, and the point x corresponding to the missing function value is substituted into the interpolation polynomial to obtain the approximate value L (x) of the missing value. Lagrange interpolation formula has a compact structure and is convenient in theoretical analysis. This method can model the missing data, which aims to be more objective and correct.

2) The original medical dataset is stored in the urban area, some attribute expressions are not matched in the same way, entity identification and attribute redundancy need to be considered. Entity recognition mainly deals with the data with different units. For example, the "first diagnostic area" in city A table and the "first diagnostic unit" in city B table of the original dataset both refer to the medical institution where the patient was first diagnosed, which can be merged into the "first diagnostic unit". Attribute redundancy mainly refers to the repeated data caused by multiple occurrences of the same attribute or inconsistent naming. This kind of data is analyzed and detected before it is deleted. For example, the meaning of the attribute "date of birth" and "age" in a dataset are the same, and the "date of birth" attribute is deleted to reduce data redundancy.

3) In the original dataset, there are data similar to ethnic and occupational dimensions. In this paper, an equal-width discretization method is adopted for processing. Within the value range of data, about 5 discrete partition points are set, and the value range is divided into some discretized intervals. For gender (male, female), treatment classification (initial treatment, diagnosis and treatment), diagnosis result (positive, negative), and other obvious attributes of the classification, the method of 0-1 unique heat coding was used to nominalize them.

4) Age attribute of the original dataset up and the range is 0 to 93, the maximum minimum normalized for such data processing. According to the proportion, and scaling to a specific area to facilitate a comprehensive analysis, the purpose is to eliminate index between the dimension and the scope of influence, the use of the normalized formulas are shown in (2) below.

$$x^* = \frac{x - \min}{\max - \min} \tag{2}$$

Where, max is the maximum age and min is the minimum age, $(\max - \min)$ is range. Deviation standardization preserves the relationship existing in the original data and is the easiest way to eliminate the influence of dimension and data value range.

C. Location Information Processing Method

It seems reasonable to use latitude and longitude to represent geographical location, but it is not suitable in the overall data set. The reasons are as follows: first, latitude and longitude are coordinate values, which cannot meet the numerical requirements of 0-1 and the accuracy is not enough; second, the purpose of location numerical is not only to convert to 0-1, but also to be related to the reality, so other attributes with strong connection are used as parameters.

Geographic location information is classified data, which needs to be processed numerically to make it easy to handle. Therefore, this paper innovatively proposes a method to highlight the differences in the weighted processing of geographic location data. The main steps are as follows:

1) Divide the geographical location data into provinces, cities, regions and counties according to the administrative region from the largest to the smallest four levels, and different initial values are set respectively. The same grade is set to the same initial value, and the initial values in order of size are set according to the classification grade.

2) A new dimension is set for the attributes that are strongly correlated with the location data, and this attribute is used as the weighted factor z. In this paper, patients are taken as the unit and the number of patients in each hospital is used as the weighted factor.

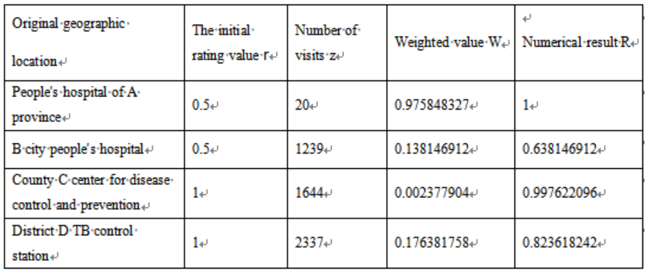
3) Use the weighting formula to calculate the value of each location data, calculate the weighted radius on the basis of the initial value and make them all within the range of 0-1.

4) Compare and analyze the results of the initial location data and the weighted data by density map, and adjust the parameters reasonably according to the density analysis. The formula for calculating weighted value is shown in (3).

$$w_i = \frac{|\ \bar{z} - z_i|}{\bar{z} + z_i},\ R_i = \begin{cases} r + w_i\,, r = 0.5 \\ r - w_i\,, r = 1.0 \end{cases} \tag{3}$$

Among them, $\bar{z}$ as the average of number, z_i is the number of patients in the i i hospital, w_i represents the weighted value, the initial value r is graded according to the administrative region, provincial and municipal hospitals set to 0.5, district and county hospitals set as 1.0, Adjust the final numerical result R_i flexibly according to w_i, in the calculation, it will be greater than 1, we unified setting it to high of 1.0. Taking four different levels of diagnosis and treatment units as an example, the main parameters are shown in Table I

TABLE I. NUMERICAL TABLE OF DIFFERENT GEOGRAPHICAL LOCATION INFORMATION



Original geographic location	The initial rating value r	Number of visits z	Weighted value W	Numerical result R
People's hospital of A province	0.5	20	0.975848327	1
B city people's hospital	0.5	1239	0.138146912	0.638146912
County C center for disease control and prevention	1	1644	0.002377904	0.997622096
District D TB control station	1	2337	0.176381758	0.823618242

The initial value is set artificially in the light of the administrative region level, so as to reflect the difference in the weighted calculation. The key point is to adjust the weight value according to the actual situation. Figure 2 shows the histogram of the numerical results of the hospital in a certain city.

Copyright (c) IARIA, 2021. ISBN: 978-1-61208-846-4

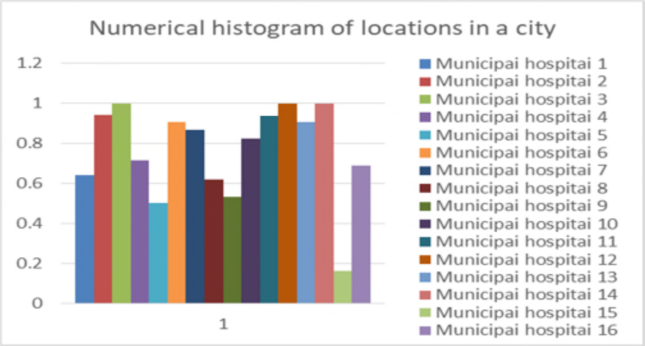


Figure. 2. Numerical histogram of locations in a city

We can see from Figure 2 that the processed position information data remains in the range of 0-1， and the standardization is realized.

II. REALIZATION OF DPC BASED ON MEDICAL DATA

In this section, we will present the Density Peak Clustering (DPC) Algorithm.

A. Density Peak Clustering Algorithm

DPC algorithm was proposed by Alex Rodriguez and Alessandro Laio [15] in 2014 and published in Science. The article "Clustering by fast search and find of density peaks" in Science mainly focuses on a kind of clustering method based on density. The main idea is to search for high-density regions separated by low-density regions. The DPC is based on the following assumptions: first, the density of the central point in a cluster is greater than that of the neighboring points; second, the distance between the center point of cluster and the higher density point is relatively large. Therefore, the DPC has two main quantities to calculate: first, the local density ρ; Second, the distance delta from the point of high density. Evidently, the center of the cluster with bigger ρ and delta relatively than others.

In order to verify that the pre-processing results of pulmonary tuberculosis medical data can be processed by machine learning, we used the popular unsupervised learning algorithm, DPC, to conduct experiments on MATLAB to extract valuable information from processed data.

B. Implementation of DPC

To reduce the time complexity and space complexity, in the original dataset, we selected the data of 11 dimensions with large influencing factors for verification: 1, the first diagnosis unit, 2, gender, 3, age 4, career, 5, treatment classification, 6, sources of patients, 7, the domicile belongs to, 8, patients diagnosis classification, 9, diagnosis, 10, reaction time, 11, diagnosis time. The decision graph is obtained by the density peak algorithm. Taking city A and city B as examples, the decision graph is shown in Figure 3 and Figure 4.

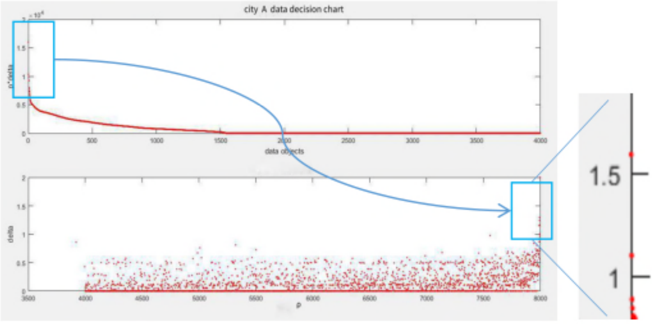


Figure. 3. Data decision chart of city A

Figure 3 is the decision diagram obtained by the density peak algorithm. The data points in the two boxes are corresponding, representing the center point in the cluster.

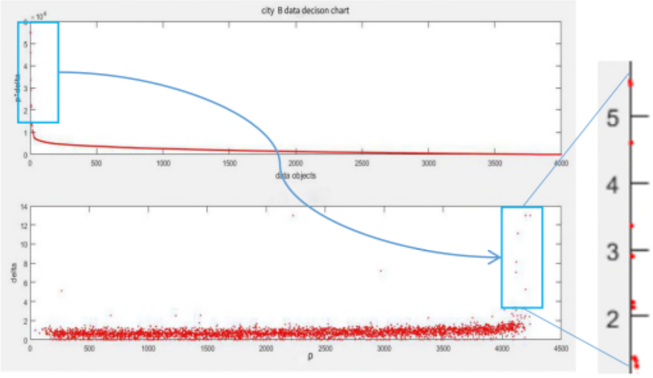


Figure. 4. Data decision chart of city B

Similar to Figure 3, cluster centers are data points with high density and high distance. It can be clearly distinguished in the decision diagram. According to the cluster center, the algorithm finds out the data in each cluster through continuous iteration and makes statistics. Figure 5 is the radar map of the first diagnosis unit after clustering in a certain city. It can be intuitively seen that among the four clusters, the two medical units with the largest number of patients are City A chronic disease prevention and treatment station, the other is B county tuberculosis prevention and treatment institute.

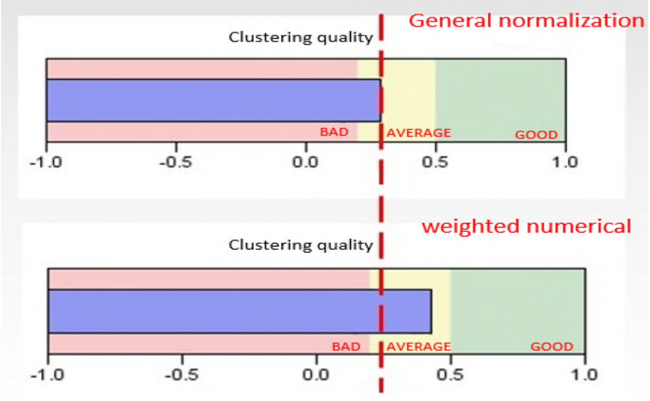


Figure. 5. Clustering quality comparison chart

By comparing the clustering quality of normalized and weighted numerical results of location attributes, it can be found that the clustering quality of innovative methods has obvious advantages from Figure 5, which also proves that the weighted numerical method is effective.

Figure 6 is the radar map of the first diagnosis unit after clustering in a certain city. It can be intuitively seen that among the four clusters, the two medical units with the largest number of patients are City A chronic disease prevention and treatment station, the other is B county tuberculosis prevention and treatment institute.

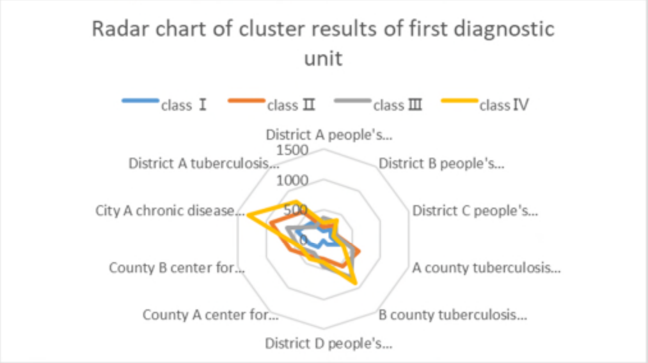


Figure. 6. Cluster radar map of the first diagnostic unit in a city

Figure 7 shows the occupational radar of patients and the reason for seeking medical treatment. It can be clearly seen that patients are mostly ordinary farmers, and they go to see a doctor after contact and infection It indicates that the pneumonia is widespread and most of them are farmers with low knowledge reserve and insufficient publicity, so that they may not be aware of it or do not know how to protect themselves from infection. In the future, knowledge discovery method will be used to analyze the statistical information of the decision graph, such as association rule algorithm, and combine the valuable attribute rules of the dataset to find the incidence rules of TB patients or treatment plans, so as to provide technical help for patients' treatment or doctors' decision making.

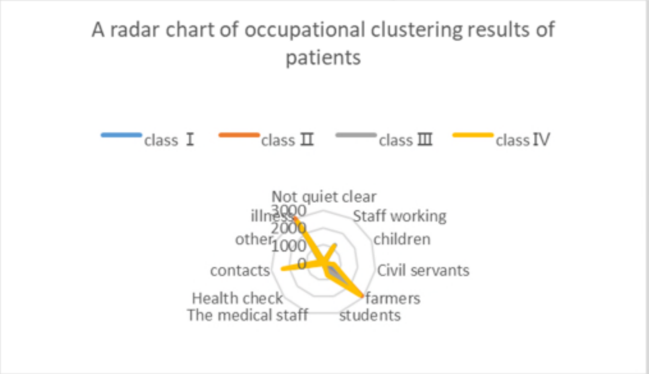


Figure. 7. Cluster radar chart of patients' occupation and reasons for seeking medical treatment in a certain city

III. CONCLUSION

In this paper, based on the treatment data of pulmonary tuberculosis patients in six urban areas, a data pre-processing method was proposed according to the characteristics of multiple attributes and non-standard. Moreover, a method to highlight the differences in the weighted processing of geographical location classification data was innovatively put forward, the original mixed datasets processing as a single numeric datasets. After that, the DPC algorithm was used to clustering data to obtain some valuable information, such as age or occupation distribution. Cluster centers and outliers were obtained by analyzing the decision graph. In future work, knowledge discovery method will be introduced, and association rules will be analyzed on the basis of clustering to discover tuberculosis treatment rules or other hidden knowledge.

ACKNOELEDGMENT

Thanks for the data support provided by Professor Li Huaichen and the respiratory team of Shandong Provincial Hospital, and thanks for the previous research foundation of Zhang Rui in the research group, which provided inspiration for the whole research and innovation.

REFERENCES

[1] Z. J Wang, S. W Mao, L. Y Yang, and P. P Tang. "A Survey of Multimedia Big Data." [J]. China Communications, vol. 15(01), pp. 155-176, 2018.

[2] M. D Li, H. Z Wang, and J. Z Li. "Mining conditional functional dependency rules on big data." Big Data Mining and Analytics vol. 3(01), pp. 68-84, 2019.

[3] G Xiang, and W Fang. "The research of Data Integration and Business Intelligent based on drilling big data." International Conference, pp. 64-68, 2017.

[4] L Zhang, Z Bing, and L Zhang. "A hybrid clustering algorithm based on missing attribute interval estimation for incomplete data." Pattern Analysis & Applications, vol. 39(1), pp.77-84, 2015.

[5] C. C Aggarwal, J Han, and R Ctr. "A framework for clustering evolving data streams." Proceedings 2003 VLDB Conference, pp.81-92, 2003.

[6] M Shih, J Jheng and L Lai. "A two-step method for clustering mixed categorical and numeric data." Tamkang Journal of Science and Engineering, vol. 13(1), pp.11-19, 2010.

[7] B. R Fatima, and J. L Diez. "Geometrical codification for clustering mixed categorical and numerical databases." [J]. Journal of Intelligent Information Systems, vol. 39(1), pp. 167-185, 2011.

[8] JM Rodríguez-Jiménez, P Cordero, M Enciso, and A Mora. "Data mining algorithms to compute mixed concepts with negative attributes: an application to breast cancer data analysis." [J]. Mathematical Methods in the Applied Sciences, pp. 4829-4845, 2016.

[9] G David, and Averbuch. "Spectral cat: categorical spectral clustering of numerical and nominal data Pattern Recognition." vol. 45(1), pp. 416-433. July 2011.

[10] Giannotti, C Gozzi, and G Manco. "Clustering Transactional Data." European Conference on Principles of Data Mining & Knowledge Discovery Springer, Berlin, Heidelberg, 2002.

[11] F Cao, J. Z Huang, and J Liang. "A fuzzy sv-k-modes algorithm for clustering categorical data with set-valued attributes." Applied Mathematics & Computation, vol. 295, pp. 1-15, Sep 2016.

[12] T Wangchamhan, S Chiewchanwattana, and K Sunat. "Efficient algorithms based on the k-means and chaotic league championship algorithm for numeric, categorical, and mixed-type data clustering." Expert Systems with Application, vol. 90(dec.30), pp. 146-167, 2017.

[13] J. Y Chen, and H. H He. "A fast density-based data stream clustering algorithm with cluster centers self-determined for mixed data." Information Sciences. pp. 271-293, 2016.

[14] S. F Ding, M. J Du, T. F Sun, X Xu, and Y Xue. "An entropy-based density peaks clustering algorithm for mixed type data employing fuzzy neighborhood." Knowledge-Based Systems, pp.133, July 2017.

[15] A Rodriguez, and A Liao. "Clustering by fast search and find of density peaks." [J]. Science, vol. 344(6191), pp. 1492-1496, 2014.