This is the published version (version of record) of:


Available from Deakin Research Online:
http://hdl.handle.net/10536/DRO/DU:30004676

Every reasonable effort has been made to ensure that permission has been obtained for items included in DRO. If you believe that your rights have been infringed by this repository, please contact drosupp@deakin.edu.au

Copyright: 2002, People's Posts & Telecommunications
Welcome Address

Dear Colleagues and Friends:

On behalf of the General Chairs of the International Conference on Intelligent Information Technology, ICIIT2002-Beijing, with great honor and pleasure, I extend to you, all the participants, and the warmest greetings.

The world is quickly stepping into the Information Age successfully on one hand, as well as problematically on the other hand. It is well recognized nowadays that Intelligent Information Technology are the keys to the Information Age. Intelligent Information Technology as the most advanced productive tools are said to be able to change human life and the world itself. However, the crisis (huge bubbles) rose from Information Technology recently clearly show that there is no such way that is straight to the Information Age. Rather, more and more questions about the Intelligent Information Technology come out in front of peoples:

- Where will the Intelligent Information Technology development to go in next few years?
- What will the Intelligent Information Technology bring about to human society?
- Why does the Information Industry, described as rising Industry by sociologists, suffer from such a great setback worldwide recently?
- Is the current Intelligent Information Technology crisis an accident happening or an unavoidable one in nature?
- Can the current Intelligent Information Technology setback be recovered quickly?
- What is the effective solution to the current Intelligent Information Technology problems?
- Why is “the gap between information has and information has not” widened instead of narrowed ever since? And so on and forth.

As scientists, professors, engineers, entrepreneurs, government officials all over the world, we have the responsibility to understand the truth and explore an effective way to a better life in Information Age. This is the motivation of ICIIT2002-Beijing.

As the host of the conferences, we will do our best to achieve the goal successfully. We wish all friends have an enjoyable stay during the conferences in Beijing.

Ruqian Lu
Co-General Chair
ICIIT 2002
Greetings from Chairs’ of Program Committee

Dear colleagues and friends:

First of all, we would like to extend to you our warmest welcome and sincere greetings on behalf of the Technical Program Committee of the International Conference on Intelligent Information Technology, ICIIT2002-Beijing.

This is the first International Conference on Intelligent Information Technology. We received over 243 papers, of which 98 papers are included in this program. We are grateful for the dedicated work of both the authors and the referees, and we hope these proceedings will continue to bear fruit over the years to come. Papers were submitted and reviewed by several referees.

A conference such as this cannot succeed without help from many individuals who contributed their valuable time and expertise. We want to express our sincere gratitude to the program committee members and referees, who invested many hours for reviews and deliberations. They provided detailed and constructive review reports that will significantly improve the papers included in the program.

We express our thanks to the Peking University Department of Computer Science that provided an effective online system to submit, track, and review the technical papers of this program.

We are very grateful to have the sponsorship of the following organizations: IEEE Computer Society, The China Computer Federation and Chinese Association of Artificial Intelligence.

We hope all of you enjoy this diverse and interesting program.

Zhongzhi Shi
Professor, Institute of Computing Technology, Chinese Academy of Sciences.

Nick Cercone,
Professor & Past Chair Computer Science
200 University Avenue
University of Waterloo
Waterloo, Ontario, CANADA

A. Cohn
University of Leeds
School of Computation Studies
United Kingdom
The Organizations of the Conferences

Organization and Program Committees

General Chairs
Benjamin W. Wah (USA)
Rujian Lu (China)
Jörg Siekmann (Germany)

Program Chairs
Zhongzhi Shi (China)
A. Cohn (England)
N. Cercone (Canada)

Program Committee Members
Z. Bubnicki (Poland)  
David W. Cheung (Hong Kong, China)  
Ming-Syan Chen (Taiwan, China)  
K. Fischer (Germany)  
Galushkin (Russian)  
Jiawei Han (USA)  
C. Jonker (Netherlands)  
Incheol Kim (Korea)  
D. Kinny (Australia)  
D. Leake (USA)  
Qing Li (Hong Kong, China)  
Pangzhen Lin (Hong Kong, China)  
Bing Liu (Singapore)  
Jian Lu (China)  
Hiroshi Motoda (Japan)  
M. Occello (France)  
Chunyi Shi (China)  
Jennie Si (USA)  
A. Smirnov (Russia)  
A. Skowron (Poland)  
Wai-Kiang Yeap (New Zealand)  
Suk I. Yoo (Korea)  
Eric Yu (Canada)  
Aidong Zhang (USA)  
Chengqi Zhang (Australia)  
Ning Zhong (Japan)  
Aoying Zhou (China)  
Hong Zhu (UK)
Referees List

Z. Bubnicki (Poland)
Qingahong Cai (China)
Enhong Chen (China)
Ming-Syan Chen (Taiwan, China)
Shifu Chen (China)
David W. Cheung (Hong Kong, China)
Jianxiang Feng (China)
K. Fischer (Germany)
Jiawei Han (USA)
Huacan He (China)
Qing He (China)
Galushkin (Russia)
C. Jonker (Netherlands)
Incheol Kim (Korea)
D. Kinny (Australia)
D. Leake (USA)
Panchang Li (China)
Jinhong Li (China)
Ning Li (China)
Qing Li (Hong Kong, China)
Weihua Li (China)
Jiaye Liang (China)
Yongguan Liang (China)
Fangzheng Lin (Hong Kong, China)
Bing Liu (Singapore)
Dayou Liu (China)
Yushu Liu (China)
Jian Lu (China)
Shaopin Ma (China)
Hiroshi Motoda (Japan)
M. Occello (France)
Lin Shang (China)
Chunyi Shi (China)
Jennie Si (USA)
A. Skowron (Poland)
A. Smirnov (Russia)
Jianping Sun (China)
Xuyan Tu (China)
Jingye Wang (China)
Changhua Wei (China)
Shize Wei (China)
Jian Wu (China)
Quanyuan Wu (China)
Fanchun Xiong (China)
Xiaofei Xu (China)
Zhuoquans Xu (China)
Anke Xue (China)
Bingru Yang (China)
Wai-Kiang Yap (New Zealand)
Suk I. Yoo (Korea)
Eric Yu (Canada)
Aidong Zhang (USA)
Chengqi Zhang (Australia)
Shijie Zhang (China)
Qingping Zhao (China)
Jinhua Zheng (China)
Ning Zhong (Japan)
Aoying Zhou (China)
Jingzhou Zhou (China)
Manli Zhou (China)
Hong Zhu (UK)
Table of Contents

Agent technology
1. Study of High Collaboration in Next Generation Open System
   Yun-young Zhang, Wei-na Guo, Jin-de Liu
   University of Electronic Science and Technology of China, China  
   1
2. A Communication Model of Mobile Agent System
   Dayou Liu, Bo Yang
   Jilin University, China  
   5
3. Initial Value Generation for Matchmaking in Middle Agents
   Zili Zhang, Chengyi Zhang
   Deakin University, Australia  
   12
4. The Architecture of Agent for Autonomous Mental Development
   Jinhua Zheng, Zhongshi Shi
   Institute of Computing Technology, CAS, China  
   19
5. Agent-based Collaboration and Parallel in Massive Spatial Information Processing
   Luo Yingwei, Wang Xiaolin, Xu Zheqiong
   Peking University, China  
   22
6. Using Reinforcement Learning to Make Automated Negotiation in Multi-Agent Based E-commerce
   Jia Li, Wang Fan, Qiu Yuhui
   Southwest China Normal University, China  
   31
7. Task Mobility in Pervasive Computing
   Jingu Zhan, Xingshe Zhou, Gang Yang
   Northwestern Polytechnical University, China  
   36
8. Agent Division as a Result of Global Optimizing Evolution
   Di Zengru, Tianwei Chen, Yougui Wang, Zhangang Han
   Beijing Normal University, China  
   40
9. Societal Abstraction for Agent-Oriented Open Giant Intelligent Systems
   Longbing Cao, Ruwei Dai
   Institute of Automation, Chinese Academy of Sciences, Beijing, China  
   47
10. A Multi-Formalism Method for Designing and Validating Intelligent Agent-Based Systems
    Michel Occello, Jean-Luc Konig, AbdelAli Ibriz, Monhammed Erradi
    CoSy Team-Leis Laboratory, France  
    53

Computer vision
1. A Method for Writer Identification Based on Off-line Handwriting
   Tianlu Mao, Jintao Li, Cong Shen, Guoqin Cui
   Institute of Computing Technology, CAS, China  
   60
2. Post-Processing for Handwritten Chinese Address Recognition
   Yuxiang Li, Xiaoyan Zhu
   Tsinghua University, China  
   64
3. Method of Visual Library of Functions Development
   Vasily V. Popovich, Yuri B. Leonov, Andrei A. Ermolenko
Knowledge discovery & Data Mining

1. A SLDNF Proof Procedure for Rule Based Update
   Sai Kiran Lakkaraju, Yan Zhang
   University of Western Sydney, Australia

2. A New Kernel Clustering Algorithm for Clustering Analysis
   Zhang Li, Weida Zhou, Licheng Jiao
   Xidian University, China

3. Re-mining Frequent Patterns Efficiently with Frequent Pattern List
   Ma Xiuli, Shiwei Tang, Dongqiang Yang, Fei Ma
   Beijing University, China

4. Towards the Semantic Web: Knowledge Representation on the Web
   Hu Yue, Li Shanping
   Zhejiang University, China

5. Eliminating Redundant Schemes and Guaranteeing Lossless Join in XML Schemes Design
   Wu Yonghui, Zhou Aoying
   Fudan University, China

Logic and logic programming

1. An Algorithm for Two Separate Tableau with Equality
   Sun Ji-Gui, Liu Quan
   Jilin University, China

2. The Verification Logic for Secure Transaction Protocols
   Qingfeng Chen, Chengqi Zhang
   Faculty of Information Technology University of Technology, Australia

3. Extending Description Logic with Default
   Dong Mingzhi, Zhang Haijun, Jiang Yuncheng, Shi Zhong Zhi
   Institute of Computing Technology, CAS, China

4. Resolution Reasoning Method in Lattice-Valued Logic LP(X)
   Jun Ma, Yang Xu, Tianrui Li
   Southwest Jiaotong University, China

Machine learning

1. A Non-iteration Fractal Coding Algorithm Based on Adaptive Block-Partition
   Ping Fu, Ping Zhong, Zhang Dongdong
   Jilin University, China

2. A New Kind of Support Vector Kernel
   Zhang Li, Weida Zhou, Licheng Jiao
7. A Statistical Method for Ridge Distance Estimation in Fingerprint Images
Yi-Long Yin, Xiao-Si Zhan, Tai-Zhe Tan, Xing-Bao Ning
Nanjing University, China
205

8. Annotation-Medical Image Annotation System Based Relevance Feedback
Lillian H Tang, Li Chen
University of Surrey, England
211

9. So You Want to Find the Canadarm... An FCMAC Approach to Configuration Estimation
Carolyn J. Schurek, Gabriele M.T. D'Eleyterlo
University of Toronto, Canada
217

10. An Improved Method of Feature Extraction from High Dimensional Data
Chen Enhong, Lin Xiao-Li
University of Science and Technology of China
225

11. Improvement and Invariance Analysis of Zernike Moments using a Region-based Shape Descriptor
Ye Bin, Peng Jia-xiong
Huazhong University of Science & Technology, P.R. China
229

Information Retrieval
1. Trademark Retrieval Based on Improved Zernike Moments Using as a Region-based Shape Descriptor
Bin-Ye, Jia-Xiong Peng
Huazhong University of Science & Technology, P.R. China
235

2. Applying Web Structure Analysis in Web Track in TREC10
Liu Yue, Yang Zhifeng, Cheng Xueqi, Wang Bin
Institute of Computing Technology, CAS, China
242

3. Personalized Intelligent Information filtering
Li Yanan, MingKai Dong, ZhiXing Huang, Yuhui Qiu
Southwest Normal University, China
248

4. Learning and Cooperating in Information Gathering
Honghui Peng, Zuoquan Lin
Peking University, China
255

5. Source Selection and Adaptation with learning in a Meta search Engine
Honghui Peng, Zuoquan Li
Peking University, China
261

Knowledge acquisition & Knowledge Representation
1. A Computable Approach to Revision of Knowledge Base
Shangmin Luan, Guozhong Dai
Institute of Software, CAS, China
269

2. A Heuristic Algorithm for Reduction of Knowledge Based on Discernibility Matrix
Duoguan Miao, Lishan Hou
Shanxi University, China
276
Evolutionary computing
1. The Evolutionary Neural Network With Tunable-activation-functions
   Shuyuan Yang, Licheng Jiao, Fang Liu
   Xidian University, China

Fuzzy & rough set
1. Learning Strategies in Robot Soccer Game Based on Fuzzy Set Rules for the Ideal Opponent Model
   Nadeem Iqbal, Lu Tian Sheng
   Shanghai Jiao Tong University, China

2. Fuzzy Modeling and Evolutionary Control of Water Supply System in Watercurtain Cooling Process
   Wang Jinhong, Qin Shi-Yin, Chen Mingyue
   Beijing Polytechnic University, China

3. Neuro-Fuzzy Modeling and Simulation for a Watercurtain Cooling Process
   Quan Jie, Qin Shou-Yin, Li Jian-Geng
   Beijing Polytechnic University, China

4. Fuzzy Approximation in Intelligent Decision Making
   Yunxiang Liu, Jigui Sun
   Jilin University, China

5. Research on the Incrementally Mining Algorithm Based on Rough Set and Its Performance
   Li Tianrui
   Southwest Jiaotong University, China

Image Processing
1. Motion Adaptive Video Format Conversion From 4:3 to 16:9 Images
   Zhang Guangjie, Zheng Nanning, Chen Qian, Wu Yong
   Xi'an Jiaotong University, China

2. Logo Recognition in Low Quality Document Images
   Ji-Kun Yan, Hui Zheng and Jian-Min Xi
   The South-West Institute of Electronics & Telecommunication Technology, China

3. Feature-Outline Based 3-D Object Localization from an Intensity Image
   Pei Mingtao, Jia Yunde, Cao Yuanda
   Beijing Institute of Technology, China

4. Medical image fusion based on CMAC
   Qu Guihong, Zhang Dali, Yan Pingfan
   Tsinghua University, China

5. Multi-dimensional Spatial Occlusion Relation
Data mining
1. The Improvement of Association Rule in Large Data Sets
   Xin Shi, Liu Shaohui
   Institute of Computing Technology, CAS, China
2. An Improved Algorithm for Feature Selection Using Fractal Dimension
   Haiqin Zhang, Qingsheng Cai
   University of Science and Technology of China
3. Symbolizing Time Series through Clustering for Classification
   Siyu Guo, Tiejun Wu
   Zhejiang University, China

Decision making
1. A Knowledge-Based Method for Modeling Collaborative Design Processes
   Tian Feng, Renhou Li
   Xi’an Jiaotong University, China
2. Adaptive Case-Based Reasoning Systems for E-Commerce
   Benchaphon Limthamnaphon, Z. Zhang and Y. Zhang
   University of Southern Queensland, Australia
3. Intention Modeling for Web Navigation
   Xiaoming Sun, Zheng Chen, Liu Wenyin, Wei-Ying Ma
   Tsinghua University, China

Distributed intelligence
1. An Intelligent Information Infrastructure for Distributed Knowledge Environments
   Rosario A. Uceda-Sosa
   IBM T.J. USA
2. Research on Real-time Interaction Oriented 3D Product Model in Distributed Collaborative Design
   Zhang Biqiang, Xing Yuan, Ruan Xueyu
   Shanghai Jiaotong University, China
3. A Distributed B-tree Based Group Key Management Scheme
   Cao Yuan-da, Wang Yong
   Beijing Institute of Technology, China

E-commerce
   Alfred Ck Heng, Ma Tit Yap
   Nanyang Technological University, Singapore
2. Improvement Approaches to Recommendation System in E-Commerce
Mobile agents & Multiagent systems

1. An Architecture of Wireless Intelligent Network with Mobile Agent
   Wang Hui, Liao Jianxin
   Beijing University of Posts and Telecommunications, China

2. Study on a SOAP-Based Communication and Service Invocation for Mobile Agents
   Wang Dan, Wei Baosheng, Yu Ge, Wang Guoren
   Northeastern University, China

3. Algorithm for Compressing the Memory Space of CMAC
   QU Guihong, ZHANG Dali, YAN Pingfan
   Tsinghua University, China

4. Modified Hierarchical Clustering Algorithm for Coordination Center Design in Multi-agent System
   Tian Chunhua, Chai Yueting, Ren Shouju, Liu Yi
   Tsinghua University, China

5. A Multi-Agent Learning Approach Based on Case and Fuzzy Theory
   Changyi Wang, Li Yao, Rizhong Wang
   National University of Defence Technology, P.R. China

6. A Planning Algorithm Based on Constraints Propagation
   Institute of Computing Technology, CAS, China
   Zhongshi Shi, Zhikun Zhao, Hu Cao

7. The Design of Intelligent Fieldbus Control and Measurement System Based on Multi-Agent Theory
   Yin Bo, Meng QingChun, Zhuang Xiaodong, Wang Hanping
   Ocean University of Qingdao, China

Multimedia and AI

1. Framework and Performance Analysis of Peer-to-Peer Intelligent Multimedia Stream Transport
Natural language processing
1. Recognition of a Kind of Words in Chinese Sign Language
   Zou Wei, Zou Wei, Yuan Kui, Liu Xianhua and Zang Aiyun
   Institute of Automatic, Chinese Academy of Sciences, China 429

Neural information processing
1. Module Based Knowledge-Increaseable Artificial Neural Network
   Qi Yingjian, Luo Siwei, Li Jianyu, Liao Lingzhi
   Northern Jiaotong University, China 435
2. A study of Adaptive Activation Function for back-propagation training
   Chien-Cheng Yu, Tsang-Min Lin
   Hsiuping Institute of Technology, China 442
3. Optimal Partition Algorithm of RBF Neural Network and Its Applications to Stock Price Prediction
   Sun Yanfeng, Zhang Wenti, Gu Xiaojiong, Liang Yan Chun, Lin Wu Zhong
   Jilin University, China 448
4. Neural Networks for Multi-Instance Learning
   Zhi-Hua Zhou, Min-Ling Zhang
   Nanjing University, P.R. China 455
5. Mining Accurate Classification Rules Using Neural Networks
   Tao Wang Qingchun Meng Tianjin Feng
   Ocean University of Qingdao, China 460

Ontology
1. Using Ontology in Agent-based Web Testing
   Qingning Huo, Hong Zhu and Sue Greenwood
   Oxford Brookes University, UK 464
2. Ontologies-Based Information Interoperability in Garment Quick Response Network
   Shi Yu, Jiaxun Chen, Fangfang Xin
   Donghua University, China 473
3. Trying to Build Ontology in Bioinformatics Using Semantic Web Language
   Qiwei Yin, Shaping Li
   Zhejiang University, China 481
4. "Web-DES" Ontology Management System
   Alexander Smirnov, Mikhail Pashkin, Nikolai Chilov, Tatiana Livashova
   St. Petersburg Institute for Information and Automation of Russian Academy of Sciences, Russian 485

Reasoning
1. Probabilistic Reasoning in the Cyclic System
   Kedian Mu, Zuoquan Lin
   Peking University, China 495
2. Reasoning about Belief Dynamics in Security Protocols
   Yan Zhang, Vijay Varadharajan
   University of Western Sydney, Australia
   502

3. An Intelligent Inference Framework for IDS Sensor Fusion
   LiHui, Xin Zan Chong, Zhao Han
   Xi'an Jiaotong University, China
   512

4. On Natural Deductive Systems for Nonmonotonic Reasoning
   Dongmo Zhang, Sai Kiran, Lakkaraju
   University of Western Sydney, Australia
   519

5. An Adaptive Negotiation Model in Multi-agent System Based on Reinforcement Learning
   Gao Yang, Ning Li, Xin Lu, Shi-fu Chen
   Nanjing University, China
   526

6. The principles and applications of artificial immune system
   Mo Hongwei, Jin Hongzhang
   Harbin Engineering University, China
   532

Text analysis
1. The Research and Implementation of Web Text Mining System
   Tang Feng, Yang Bingru
   Information Engineering School, China
   536

   Bangyong Liang, Jianjun Xu, Juanzi Li, Kehong Wang
   Tsinghua University, China
   540

3. Efficient Index Structure for Similarity Search in Large Metric Spaces
   Zhaogong Zhang, Jian Zhong Li
   Department of Computer Science, China
   545

4. Special Topic Organization and Retrieval System
   Ziyan Jia, Qing He, Haijun Zhang, Zhongzhi Shi
   Institute of Computing Technology, CAS, China
   552

Neural information processing
1. Classifying and Learning Based on Qualitative Mapping
   Wang Lihu, Feng Jialai, Li Yu, Sheng Zhaohan, Dong Zhangfuj
   Graduate School of Management Science & Engineering Nanjing University, China
   559

Agent
1. Dynamic Contract Net Protocol
   Zhang Haijun, Wu Bin, Dong Mingkai, Shi Zhongzhi
   564

2. Information Gain and Discovery Model Based on MultiAgent System
   LUO Xiangfeng, ZHENG Shuli
   573

Authors Index

viii
Initial Value Generation for Matchmaking in Middle Agents

Zili Zhang
School of Computing and Mathematics
Deakin University
Geelong Victoria 3217, Australia
email: zzhang@deakin.edu.au

Chengqi Zhang
Faculty of Information Technology
University of Technology, Sydney
PO Box 123 Broadway, NSW 2007 Australia

Yuefeng Li
School of Software Engineering and Data Communications
Queensland University of Technology
Brisbane Queensland 4001, Australia

Abstract

One of the major challenges that agents used in open environments must face is that they must be able to find each other. This is because in an open environment, agents might appear and disappear unpredictably. To address this issue, middle agents have been proposed. The performance of middle agents relies heavily on the matchmaking algorithms used. Matchmaking is the process of finding an appropriate provider for a requester through a middle agent. The practical performance of service provider agents has a significant impact on the matchmaking outcomes of middle agents. Thus the track records of agents in accomplishing similar tasks in the past should be taken into account in matchmaking process. Considering that there are no track records available at the launching of an agent system, this paper discusses some ways to provide reasonable initial values for the track records. With the agents' history and the initial values of the track records, the performance of matchmaking algorithms can be improved significantly.

1. Introduction

An agent is an encapsulated computer system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives [1]. A multi-agent system can be defined as a loosely coupled network of agents that work together to make decisions or solve problems that are beyond the individual capabilities or knowledge of each agent [2].

One of the most important driving forces behind multi-agent system research and development is the Internet: agents are populating the Internet at an increasingly rapid pace. These agents invariably need to interact with one another in order to meet their designer's objectives. In such open environments, agents that would like to coordinate with each other (either cooperate or negotiate, for example) face two major challenges: first, they must be able to find each other (in an open environment, agents might appear and disappear unpredictably), and once they have done that, they must be able to inter-operate [3]. To address the issue of finding agents in an open environment like the Internet, middle agents [4] have been proposed. Each agent advertises its capability to some middle agent. A number of different agent types have been identified, including matchmakers or yellow page agents (that match advertisements to requests for advertised capabilities), blackboard agents (that collect requests), and brokers (that process both). Middle agents are advantageous since they allow a system to operate robustly in the face of agent appearance and disappearance, and intermittent communications.

The performance of middle agents relies heavily on the matchmaking algorithms used. Matchmaking is the process of finding an appropriate provider for a requester through a middle agent. The state of the art of matchmaking algorithms is as follows: Agents provide meta-information about the services they offer to middle agents. Based on relevance calculation methods (mainly distance functions) the middle agents use this information for matching and provide the best fitting offer to a requester agent. The fact that is not taken into account is that the meta-information is defined by the provider agent itself. This means that in general the advertisements of the provider agents are not proven but self defined.

On the other hand, there are usually more than one service provider agents claim that they have the same or very similar capabilities to accomplish a task in an application.
For example, in financial investment applications one usually needs to predict the interest rate. There are different techniques for interest rate prediction such as neural network (NN) technique and fuzzy logic with genetic algorithm (FLGA) [6], but their prediction performances are different. If there are two interest rate prediction agents, one based on NN, the other based on FLGA, which one should be chosen to predict the interest rate? In such cases, current matchmaking algorithms can only choose one provider agent randomly. As the quality of service of different service provider agents varies from one agent to another, even though they claimed they have the same capabilities, it is obvious that it is not a good strategy to randomly choose a provider agent for the requester agents.

To this end, we provided a solution to this issue by introducing an algorithm which can consider agents’ track records [5]. The work described in [5] advances the state of the art by enhancing the existing matchmaking approaches by a quality measure which relates to the agents performance in live applications.

The track records or “credit histories” of agents are accumulated gradually during the executing process of a multi-agent system. Thus there are no track records available when the system is first launched. In such cases the improved matchmaking algorithm can only randomly choose one agent with the requested capability, which is the same as other matchmaking algorithms.

If one can find some way to provide reasonable initial values for the track records, one can choose the agents based on the initial values at the very beginning. With the track records accumulated, the initial values and the track records are combined to choose the agents. In this way, the shortcoming of the algorithm considering track records – no track records being available at the very beginning – is overcome. This is just the task of this paper.

This paper will propose some ways to provide initial values for the track records. The basic idea is to give a set of problems (called benchmark problems), and ask all the agents who claimed that they have the capabilities to solve these problems. By calculating the “distances” between the solutions to the problems the agents gave and the benchmark results, one can determine the initial values for the track record of these agents. Of course, if the benchmark results are unknown in advance, we must find the benchmark results first.

The rest of the paper is structured as follows. Section 2 gives the basic idea for initial value generation of track records. Section 3 is the description of benchmark problems and benchmark values. Section 4 presents the initial value generation process with known benchmark results. When benchmark results are unknown, an approach is proposed to find them in Section 5. Section 6 concludes the paper.

2. Basic Idea for Initial Value Generation

Suppose we want to delegate a task to a person, but we have ten persons all claimed that they have the capability to accomplish the task. In such a case whom should we choose to delegate the task? If we have the track records of the ten persons in accomplishing similar tasks in the past, we can make the decision based on their performance/accomplishment history. This is what we discussed in [5]. If there is no any information about the ten persons in accomplishing similar tasks, one simple and efficient way we usually utilize in real life is to design a set of problems and ask all the candidates to solve these problems. We assess the solutions provided by the candidates and delegate the task to the candidate with the best solutions. The simplest form of such kinds of problems is different examination papers. In this paper we call such a set of problems “benchmark problems”. The “standard” solutions are called “benchmark results” or “benchmark values”.

According to the above scenario, we can summarize the basic idea of initial value generation approach as follows. Before putting a multi-agent system into practical operation, the system is “trained” with a set of benchmark problems. That is, the middle agent is run with a matchmaking algorithm first (e.g., find.nn algorithm [7]). The middle agent then asks the agents with the same or similar capabilities (based on the returned results of the matchmaking algorithm such as find.nn) to solve the benchmark problems. By testing the results provided by these agents against the benchmarks, one obtains an evaluation of these agents for their performance on solving the benchmark problems. This evaluation is then used as the initial value of these agents’ track records.

3. Description of Benchmark Problems and Benchmark Values

In order to extract an appropriate description of benchmark problems and benchmark values, we take the software risk analysis as an example.

Assume there is a software project and we want to analyze the risk of this project. From software engineering risk management point of view, there are some principal software risk factors that influence the risk of a software project [8]. These software risk factors include organization, estimation, monitoring, development methodology, tools, risk culture, usability, correctness, reliability, and personnel. The software risk factor organization addresses risks associated with the maturity of the organization structure, communications, functions, and leadership; The software risk factor estimation focuses on risks associated with inaccurate estimations of the resources, schedules, and costs.
Table 1. Description of Benchmark (BM) Problems

<table>
<thead>
<tr>
<th>Agent</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$\ldots$</th>
<th>$S_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>$a_1$</td>
<td>$a_2$</td>
<td>$\ldots$</td>
<td>$a_m$</td>
</tr>
<tr>
<td>$A_2$</td>
<td>$b_1$</td>
<td>$b_2$</td>
<td>$\ldots$</td>
<td>$b_m$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\ddots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$A_k$</td>
<td>$a_{1k}$</td>
<td>$a_{2k}$</td>
<td>$\ldots$</td>
<td>$a_{mk}$</td>
</tr>
<tr>
<td>BM</td>
<td>$b_1$</td>
<td>$b_2$</td>
<td>$\ldots$</td>
<td>$b_m$</td>
</tr>
</tbody>
</table>

eeded to develop software; The software risk factor monitoring refers to risks associated with identifying problems; The software risk factor development methodology identifies the methods by which software is developed; The software risk factor tools focus on risks associated with the software tools used when software is developed; and so on. To analyze the risk of a software project is to determine the values (e.g., low, medium, high etc.) of these software risk factors. The software risk factors here can be viewed as attributes describing the software risk problem. More generally, we can say that solving a problem is to find the attribute values related to the problem.

Formally, let $A = \{A_1, A_2, \ldots, A_k\}$ be the agent set with the same or similar capabilities. We use $S = \{S_1, S_2, \ldots, S_m\}$ to denote the problem set. Each problem $S_i \in S$ ($i = 1, 2, \ldots, m$) has a related attribute set $a = \{a_1, a_2, \ldots, a_m\}$. We say agent $A_i$ solved problem $S_j$ if it returns the values of the $n$ attributes related to the problem. The values can be numeric or non-numeric (linguistic values). Suppose the benchmark values of these attributes denoted by $B_i = \{b_{i1}, b_{i2}, \ldots, b_{im}\}$ and the values returned by agent $A_j$ denoted by $B_{ij} = \{b_{1j}, b_{2j}, \ldots, b_{mj}\}$ ($i = 1, 2, \ldots, m, j = 1, 2, \ldots, k$). The description of the benchmark problems is then summarized in Table 1.

The next step in the initial value generation process is to calculate the “distances” between the returned values by agent $A_j$ and the benchmark values. There are many definitions of “distance.” Here the distance is defined in terms of standard Euclidean distance. The distance between $B_i$ and $B_{ij}$ is defined to be $d_j$, where $d_j = \sqrt{\sum_{i=1}^{m} (b_{ij} - b_{ij})^2}$. Then these distances are added to the database of the middle agent as the initial values of the track records.

Considering that the initial values and the track records need to be combined when accumulated, the distances were mapped to the satisfactory degrees defined in [5]. As there are 7 levels of satisfactory degrees—strong satisfaction, satisfaction, weak satisfaction, neutral, weak unsatisfaction, unsatisfaction, and strong unsatification, each level counts for 1/7 of the distance range. Therefore, if the distance is between 0 and 0.143, strong satisfaction will be the initial value of the agent’s track record; if the distance is between 0.143 and 0.286, satisfaction will be the initial value etc. The mapping results are shown in Table 2.

Table 2. Mapping Results between Distance and Satisfactory Degree

<table>
<thead>
<tr>
<th>Distance Range</th>
<th>Satisfactory Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 0.143</td>
<td>strong satisfaction</td>
</tr>
<tr>
<td>0.143 to 0.286</td>
<td>satisfaction</td>
</tr>
<tr>
<td>0.286 to 0.429</td>
<td>weak satisfaction</td>
</tr>
<tr>
<td>0.429 to 0.572</td>
<td>neutral</td>
</tr>
<tr>
<td>0.572 to 0.715</td>
<td>weak unsatisfaction</td>
</tr>
<tr>
<td>0.715 to 0.858</td>
<td>unsatisfaction</td>
</tr>
<tr>
<td>0.858 to 1.0</td>
<td>strong unsatisfaction</td>
</tr>
</tbody>
</table>

In the process of initial value generation, there are two situations that need to be considered. One is the benchmark values in Table 1 are known in advance, the other is the benchmark values are unknown. We will discuss these cases respectively in the subsequent sections with examples.

4. Initial Value Generation with Known Benchmark Results

For different applications, the benchmark problems are different. That is, the benchmark problems are application-dependent. In this section, we take the financial application as an example to discuss the initial value generation problem with known benchmark values.

In financial applications, different models (e.g., fuzzy logic and genetic algorithm model [6]) can be used for interest rate prediction. In this section, two soft computing (SC) agents for interest rate prediction (one is based on neural network, called SC-Agent_NN, the other based on fuzzy logic and genetic algorithm, called SC-Agent_FLGA) are taken as examples to show how to determine the initial values for the two agents. The initial values are based on the predictive capabilities of these two SC agents.

4.1. Construction of Benchmark Problems

When predicting the interest rate (as represented by 91-day Treasury bill rates), both of the agents take the changes of previous Treasury—bill (T—bill) rates, real gross national product (GNP), consumer price index (CPI), M2 money supply, and personal wealth (W) as inputs. Personal wealth is the accumulation of the difference between personal income and personal consumption. The M2 money supply consists of all cash in circulation and deposits in savings.
and check accounts, and represents readily available liquid assets. The consumer price index is a measure of the inflation trend. The outputs are the changes of next T-bill rates (predicted interest rates). Quarterly data are used.

We use the history financial data of the five factors (from 1966 to 1987) provided in Appendix B of [6] to construct the required benchmark problems.

There is some evidence to suggest that fundamental financial market characteristics change over a period of four to five years [9]. That is, the market "forgets" the influence of data that is more than five years old. For this reason, five-year data windows are used. 15 data windows are examined, each starting in the first quarter of the years 1967 through to 1981, respectively. The ending quarters for each data window will be the fourth quarters of the years 1971 through to 1985. This means there are 15 benchmark problems. The inputs of benchmark problem $S_1$, for example, are the data from 1967 to 1971. The benchmark value for these inputs is the T-bill rate of the first quarter of 1972, -0.81.

4.2. Experimental Results

The 15 data windows are used to train these two agents (neural network and genetic algorithm). We then let the agents predict the interest rate of the first quarter following the training data windows. For example, for training data of 1967-1971, the outputs of the agents are the (predicted) T-bill rate of the first quarter of 1972. The prediction results of the two agents on the 15 benchmark problems are summarized in Table 3.

The average distance for the prediction values of SC.Agent_NN is 0.287. The value for SC.Agent_FIGA is 0.123. Based on the prediction results and the average distances, one can see that the prediction performance of SC.Agent_FIGA is better than that of SC.Agent_NN for the benchmark problems. Mapping the distances to satisfactory degrees (refer to Table 2), weak satisfaction is added to the track record of SC.Agent_NN as its initial value, and strong satisfaction to the track record of SC.Agent_FIGA as its initial value. Hence at this stage, if the middle agent needs to pick one agent for interest rate prediction, the result is SC.Agent_FIGA. Of course, the situation may change with the accumulation of track records of these agents.

5. Initial Value Generation with Unknown Benchmark Results

In the case discussed in the previous section, one must know a priori of the attribute values of the benchmark problems, but this is not always the case. In some situations, it is impossible to obtain the attribute values in advance. In such cases, one can ask the agents to solve these problems first. One can then try to cluster the attribute values returned by agents using cluster analysis methods. In this way, "heuristic" attribute values can be obtained. One can then use the "heuristic" attribute values as the benchmark values and back to the situation discussed in the previous section.

There are seven steps involved using cluster analysis algorithms to determine the benchmark results. Before presenting these steps in details, a brief introduction to fuzzy cluster analysis algorithms [10][11], which are used in our experiments, is given.

5.1. A Brief Introduction to Fuzzy Cluster Analysis

The aim of a cluster analysis is to partition a given set of data or objects into clusters (prototypes). This partition should have the following properties: (1) Homogeneity within the cluster, i.e., data that belong to the same cluster should be as similar as possible; (2) Heterogeneity between clusters, i.e., data that belong to different clusters should be as different as possible. The concept of "similarity" has to be specified according to the data. Since the data are in most cases real-valued vectors, the Euclidean distance between data can be used as a measure of the dissimilarity.

The fuzzy clustering algorithms that we will use classify the elements of the data set $X = \{x_1, x_2, \ldots, x_j, \ldots, x_n\} \subset \chi$ into classes $P = \{P_1, P_2, \ldots, P_i, \ldots, P_c\} \subset \varphi$ by means of a membership matrix $U \subset [0,1]^{c \times |X|}$. A membership grade $u_{ij}$ denotes the degree of belongingness of datum $x_j$ to class $P_i$. The algorithms minimize the following objective function by means of alternating optimization (AO)

$$J(X, W; U, P) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^\mu w_{ij} d^\delta(k_i, x_j),$$

where $d : \chi \times \varphi \rightarrow R$ measures the distance between data vectors and prototypes and $w_{ij}$ is the weight for the data vector $x_j$. The term alternating optimization comes from the fact that $J$ is minimized by updating prototypes and membership alternatively. If $X = \{x_1, x_2, \ldots, x_j, \ldots, x_n\}$ and $P = \{P_1, P_2, \ldots, P_i, \ldots, P_c\}$ are finite, an analysis result $f : X \rightarrow F(P)$ can be represented as a $c \times n$ matrix $U$, where $u_{ij} := f(x_j)(P_i)$. The algorithm used is shown below.

Fuzzy Clustering Algorithm

Let a data set $X = \{x_1, x_2, \ldots, x_j, \ldots, x_n\}$ be given. Let each cluster be uniquely characterizable by an element of a set $P$.

Choose the number of clusters, $2 \leq c \leq n$

Choose an $m \in R$ (m > 1)

Choose a precision for termination s
Table 3. Predicting Results on Benchmark Problems

<table>
<thead>
<tr>
<th>Agent</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
<th>$S_5$</th>
<th>$S_6$</th>
<th>$S_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC.Agent_NN</td>
<td>-0.53</td>
<td>0.45</td>
<td>0.03</td>
<td>-0.93</td>
<td>-0.67</td>
<td>-0.14</td>
<td>0.43</td>
</tr>
<tr>
<td>SC.Agent_FG</td>
<td>-0.78</td>
<td>0.73</td>
<td>-0.15</td>
<td>-1.36</td>
<td>-0.60</td>
<td>-0.28</td>
<td>0.53</td>
</tr>
<tr>
<td>Benchmark</td>
<td>-0.81</td>
<td>0.70</td>
<td>-0.40</td>
<td>-1.20</td>
<td>-0.69</td>
<td>-0.28</td>
<td>0.47</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$S_8$</th>
<th>$S_9$</th>
<th>$S_{10}$</th>
<th>$S_{11}$</th>
<th>$S_{12}$</th>
<th>$S_{13}$</th>
<th>$S_{14}$</th>
<th>$S_{15}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.79</td>
<td>1.41</td>
<td>2.15</td>
<td>-0.95</td>
<td>-1.16</td>
<td>-0.33</td>
<td>-0.45</td>
<td>-0.41</td>
</tr>
<tr>
<td>1.03</td>
<td>1.84</td>
<td>2.66</td>
<td>-1.17</td>
<td>-1.00</td>
<td>-0.22</td>
<td>-0.87</td>
<td>-0.53</td>
</tr>
<tr>
<td>1.02</td>
<td>1.91</td>
<td>2.57</td>
<td>-1.10</td>
<td>-0.82</td>
<td>-0.03</td>
<td>-1.08</td>
<td>-0.11</td>
</tr>
</tbody>
</table>

Initialize $U^{(0)}$, $i := 0$.

REPEAT

Increase $i$ by 1.

Determine $P^{(i)}$ such that $J$ is minimized by $P^{(i)}$ for fixed $U^{(i-1)}$.

Determine $U^{(i)}$ according to certain conditions

UNTIL $\|U^{(i-1)} - U^{(i)}\| \leq \varepsilon$

There are many ways to determine $U^{(i)}$. Different ways result in different fuzzy clustering algorithms. Refer to [10] for more details.

In the next subsection, we will discuss the steps in determining benchmark values by using this fuzzy clustering algorithm.

5.2. Determining Benchmark Values by Fuzzy Clustering

As discussed in Section 3, finding a solution to any problem can be viewed as determining the attribute values related to the problem. With this observation in mind, a set of benchmark problems can be designed, but the solutions to all these problems are unknown (e.g., we do not know what the risk exactly is for a software project). Some agents who claimed to have the capabilities are asked to solve these problems and return the solutions (the attribute values). These attribute values are then used as inputs to the fuzzy clustering algorithm. The algorithm partitions the solutions into different clusters (called prototypes). We then choose the center of the cluster with the highest weight as the benchmark values for the problems. This implies that we should accept most agents' opinions if their solutions are similar. It is reasonable to do so. Specifically, this process involves the following steps.

- Preparation: Given $k$ similar problems (benchmark problems). Choose $m$ agents having the capabilities to solve the $k$ problems, respectively. The solution of each problem consists of $n$ attributes.

- Use the fuzzy clustering algorithm to determine the clusters (prototypes) of the solutions for each of the $k$ problems returned by the $m$ agents. That is, the input of the algorithm is $m \times n$ data matrix. There are $k$ such matrices. The outputs of the algorithm have the following format:

  (cluster)
  (prototype (weight value) (center (values for all the attributes)) ...)

(prototype (weight value) (center (values for all the attributes))

- Choose the prototype with the highest weight as the benchmark values for this problem. Repeat this step for $k$ problems.

- Calculate the distances (using Euclidean distance) of the solutions given by the agents and the found benchmark values.

- Find the average distances of all the solutions provided by the $m$ agents with the found benchmark values of the $k$ problems.

- Normalize the distances to $[0, 1]$. Sort the $m$ agents according to the average distances. Mapping the average distances to the seven satisfaction degrees (refer to Table 2), and use the satisfaction degrees as the initial values of these agents, respectively.

The average distances are used as the measurement of agents' performance in accomplishing benchmark problems. Such a measurement meets the "majority principle". That is, if one agent can accomplish most of the benchmark problems with high quality, another agent can only accomplish a few of the benchmark problems with high quality, the average distance of the first agent is shorter than that of the second one. Therefore, the results obtained according to the above process are convincing. The key in this process is step 2. In this step fuzzy clustering algorithm is used to cluster the solutions provided by different agents. Based on the clustering results, the "heuristic" attribute values, which are reasonable benchmark values, are determined.
5.3. Experimental Results

Recall the software project risk example in Section 3. Suppose some experts in this field are invited to assess the risks of some software projects. The experts are asked to give their assessment results by providing a number (e.g., [0, 10]) for each of the ten software risk factors. The higher the number, the higher the risk concerning that risk factor. According to the process describing in the previous subsection, some experiments were conducted with k = 30, m = 20, and n = 10. That is, 20 agents with similar capabilities are delegated to assess 30 software projects (benchmark problems). The answer for each problem consists of 10 attributes. The data for the risk factors used in the experiments are randomly generated. For demonstration purpose, Table 4 shows the solutions of the 10 agents for one of the 30 benchmark problems.

Taking this as inputs, the clustering algorithm produces the following output for this problem:

<table>
<thead>
<tr>
<th>cluster</th>
<th>prototype(weight)</th>
<th>center</th>
<th>(7.052511, 8.4176, 3.82815, 5.14071)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>(9.04928, 1.94891, 1.56994, 0.253258, 4.76614, 4.21243)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(2.91169, 5.33178, 5.17589, 2.66183, 2.15821, 8.16924, 7.56102, 7.51877, 3.92546)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(4.545996, 6.99357, 1.95553, 3.98165, 5.02628, 3.01206, 2.02627, 7.99989, 6.01413, 5.01984, 3.98599)</td>
</tr>
</tbody>
</table>

There are three different clusters (prototypes). Choosing the prototype with the highest weight (5.45996), we obtain the benchmark values for this problem. The attribute values are $a_1 = 6.99357$, $a_2 = 1.95553$, $a_3 = 3.98165$, $a_4 = 5.02628$, $a_5 = 3.01206$, $a_6 = 2.02627$, $a_7 = 7.99989$, $a_8 = 6.01413$, $a_9 = 5.01984$, and $a_{10} = 3.98599$, respectively. Table 5 lists the benchmark values for the first 5 benchmark problems determined by fuzzy clustering approach.

We then calculate the Euclidean distances between the solutions provided by the agents and the found benchmark values. The results are shown in Table 6. As space is limited, only the distances between 5 agents’ solutions and benchmark results of 5 benchmark problems are listed.

To measure the performance in accomplishing benchmark problems, the average distances between agents’ solutions and the benchmark values of all benchmark problems are used. The shorter the average distance, the better the performance. The average distance between agent $A_i$ and the benchmark values is denoted by $d_{Ai}$. Based on the experimental data, these distances are $d_{A_1} = 0.2141$, $d_{A_2} = 0.0631$, $d_{A_3} = 0.0482$, $d_{A_4} = 0.0941$, $d_{A_5} = 0.1795$, $d_{A_6} = 0.3442$, $d_{A_7} = 0.2591$, $d_{A_8} = 0.3137$, $d_{A_9} = 0.1824$, $d_{A_{10}} = 0.2666$, $d_{A_{11}} = 0.2181$, $d_{A_{12}} = 0.215$, $d_{A_{13}} = 0.3946$, $d_{A_{14}} = 0.0777$, $d_{A_{15}} = 0.1818$, $d_{A_{16}} = 0.1333$, $d_{A_{17}} = 0.2724$, $d_{A_{18}} = 0.2097$, $d_{A_{19}} = 0.1442$, $d_{A_{20}} = 0.4322$, respectively. From the average distances, it is obvious that agent 3 has the best performance in doing the benchmark problems, agent 2 has the second best performance, etc. Mapping these average distances to satisfactory degrees according to Table 2, the initial values of agents 2, 3, 4, 14, and 16 are strong satisfactions as their average distances are within (10, 0.143); the initial values of agents 1, 5, 7, 9, 10, 11, 12, 15, 18, and 19 are satisfactions as their average distances are between 0.143 and 0.286; the initial values of agents 6, 8, 13, and 17 are weak satisfactions; and the initial value of agent 20 is neutral.

6. Conclusions

Matchmaking in middle agents is essential for multi-agent systems used in open environments such as the Internet. Agents’ track records have a strong impact on the outcome of matchmaking. Therefore agents’ history performance/accomplishment (track records) should be taken into account in matchmaking. As the track records of agents are accumulated gradually during the executing process of a multi-agent system, there are no track records available when the system is first launched. To this end, this paper proposed ways to generate reasonable initial values for track records of agents.

The basic idea for initial value generation is to provide a set of benchmark problems and ask all provider agents claimed to have the same capabilities to solve these problems. By comparing the distances between the solutions provided by agents and the benchmark values, the initial values of these provider agents are then determined.

Two cases were identified in initial value generation. If the benchmark values are known in advance, the distances were calculated directly to determine the initial values of agents. If the benchmark values are unknown, fuzzy clustering algorithms were employed to find the benchmark values first. In both situations, experiments were conducted. The experimental results show the proposed initial value generation approaches are workable and can produce reasonable initial values. Combining the agents’ history performance/accomplishment information and the initial values of track records, the performance of matchmaking algorithms can be improved significantly.

Thus far, all the discussions are based on one assumption: The track records of agents are credible. The situation with false track records is subject to further research.

Acknowledgement

The authors would like to thank Mr Hong Hu’s support in conducting the experiments.

References

Table 4. Agents’ Solutions to One Benchmark Problem

<table>
<thead>
<tr>
<th>Agent</th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$a_4$</th>
<th>$a_5$</th>
<th>$a_6$</th>
<th>$a_7$</th>
<th>$a_8$</th>
<th>$a_9$</th>
<th>$a_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>6.94</td>
<td>1.85</td>
<td>3.89</td>
<td>5.38</td>
<td>3.24</td>
<td>1.99</td>
<td>8.21</td>
<td>5.84</td>
<td>5.02</td>
<td>3.85</td>
</tr>
<tr>
<td>$A_2$</td>
<td>6.93</td>
<td>2.04</td>
<td>4.07</td>
<td>4.96</td>
<td>2.97</td>
<td>1.95</td>
<td>8.06</td>
<td>6.0</td>
<td>4.97</td>
<td>3.97</td>
</tr>
<tr>
<td>$A_3$</td>
<td>6.99</td>
<td>1.99</td>
<td>4.0</td>
<td>5.0</td>
<td>3.0</td>
<td>2.0</td>
<td>8.0</td>
<td>5.99</td>
<td>5.0</td>
<td>4.01</td>
</tr>
<tr>
<td>$A_4$</td>
<td>7.12</td>
<td>1.9</td>
<td>4.08</td>
<td>4.87</td>
<td>3.14</td>
<td>2.01</td>
<td>8.07</td>
<td>5.99</td>
<td>5.15</td>
<td>3.87</td>
</tr>
<tr>
<td>$A_5$</td>
<td>7.06</td>
<td>1.93</td>
<td>4.22</td>
<td>5.21</td>
<td>3.31</td>
<td>2.14</td>
<td>8.14</td>
<td>5.76</td>
<td>4.88</td>
<td>3.8</td>
</tr>
<tr>
<td>$A_6$</td>
<td>7.15</td>
<td>1.65</td>
<td>3.65</td>
<td>4.32</td>
<td>3.42</td>
<td>2.01</td>
<td>7.66</td>
<td>6.03</td>
<td>5.16</td>
<td>3.57</td>
</tr>
<tr>
<td>$A_7$</td>
<td>6.7</td>
<td>2.41</td>
<td>4.44</td>
<td>5.14</td>
<td>2.65</td>
<td>2.27</td>
<td>8.42</td>
<td>5.54</td>
<td>4.9</td>
<td>3.8</td>
</tr>
<tr>
<td>$A_8$</td>
<td>6.93</td>
<td>2.03</td>
<td>3.69</td>
<td>5.28</td>
<td>2.99</td>
<td>1.75</td>
<td>8.45</td>
<td>6.46</td>
<td>4.45</td>
<td>4.34</td>
</tr>
<tr>
<td>$A_9$</td>
<td>6.86</td>
<td>1.9</td>
<td>3.73</td>
<td>4.78</td>
<td>3.21</td>
<td>2.15</td>
<td>7.87</td>
<td>6.02</td>
<td>5.17</td>
<td>4.29</td>
</tr>
<tr>
<td>$A_{10}$</td>
<td>6.93</td>
<td>1.62</td>
<td>3.68</td>
<td>4.98</td>
<td>2.88</td>
<td>1.74</td>
<td>8.13</td>
<td>6.39</td>
<td>4.97</td>
<td>4.38</td>
</tr>
</tbody>
</table>

Table 5. Partial Benchmark Values for Benchmark Problems

<table>
<thead>
<tr>
<th>Problems</th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$a_4$</th>
<th>$a_5$</th>
<th>$a_6$</th>
<th>$a_7$</th>
<th>$a_8$</th>
<th>$a_9$</th>
<th>$a_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_4$</td>
<td>2.888</td>
<td>2.998</td>
<td>8.035</td>
<td>7.036</td>
<td>8.002</td>
<td>2.001</td>
<td>4.992</td>
<td>3.013</td>
<td>8.069</td>
<td>9.009</td>
</tr>
<tr>
<td>$P_5$</td>
<td>7.983</td>
<td>3.922</td>
<td>7.071</td>
<td>7.013</td>
<td>1.968</td>
<td>7.041</td>
<td>1.995</td>
<td>3.988</td>
<td>7.017</td>
<td>5.000</td>
</tr>
</tbody>
</table>

Table 6. Distances between Agents’ Solutions and Benchmark Values

<table>
<thead>
<tr>
<th>Agents</th>
<th>$P_1$</th>
<th>$P_2$</th>
<th>$P_3$</th>
<th>$P_4$</th>
<th>$P_5$</th>
<th>$P_6$</th>
<th>$P_7$</th>
<th>$P_8$</th>
<th>$P_9$</th>
<th>$P_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>0.2148</td>
<td>0.2611</td>
<td>0.2258</td>
<td>0.2229</td>
<td>0.2522</td>
<td>0.1715</td>
<td>0.1722</td>
<td>0.1899</td>
<td>0.2533</td>
<td>0.1963</td>
</tr>
<tr>
<td>$A_2$</td>
<td>0.0554</td>
<td>0.0389</td>
<td>0.0622</td>
<td>0.0749</td>
<td>0.0403</td>
<td>0.0654</td>
<td>0.0579</td>
<td>0.0612</td>
<td>0.0509</td>
<td>0.0385</td>
</tr>
<tr>
<td>$A_3$</td>
<td>0.0467</td>
<td>0.0364</td>
<td>0.0486</td>
<td>0.0428</td>
<td>0.0394</td>
<td>0.0435</td>
<td>0.0559</td>
<td>0.0342</td>
<td>0.0341</td>
<td>0.0303</td>
</tr>
<tr>
<td>$A_4$</td>
<td>0.0795</td>
<td>0.1035</td>
<td>0.0904</td>
<td>0.088</td>
<td>0.0813</td>
<td>0.0899</td>
<td>0.072</td>
<td>0.0831</td>
<td>0.1101</td>
<td>0.0627</td>
</tr>
<tr>
<td>$A_5$</td>
<td>0.1604</td>
<td>0.2026</td>
<td>0.1785</td>
<td>0.1585</td>
<td>0.1788</td>
<td>0.1578</td>
<td>0.1864</td>
<td>0.1554</td>
<td>0.2265</td>
<td></td>
</tr>
</tbody>
</table>

1997, 26-37.


