Argyros, A.; Bártfai, G.; Eitzinger, Ch.; Kemény, Zs.; Csáji, B. Cs.; Kék, L.; Lourakis, M.; Reisner, W.; Sandrisser, W.; Sarmis, T.; Umgeher, G.; Viharos, Zs. J.; Smart Sensor Based Vision System for Automated Processes, International scientific book Emerging Technologies, Robotics and Control Systems, International Society for Advanced Research, Thomson Scientific Journal, ISBN: 978-88-901928-9-5, 2007, pp. 24-29.

Smart Sensor Based Vision System for Automated Processes

Antonis Argyros[i]; Gusztáv Bártfai[v]; Christian Eitzinger[iv]; Zsolt Kemény[ii]; Balázs Csanád Csáji[ii]; László Kék[ii,iii]; Manolis Lourakis[i]; Werner Reisner[iv]; Wolfgang Sandrisser[vi]; Thomas Sarmis[i]; Gerald Umgeher[iv]; Zsolt János Viharos[ii];

[i]) Foundation for Research and Technology–Hellas, Heraklion, GREECE argyros@ics.forth.gr

[ii]) Computer and Automation Research Insitute, Hungarian Academy of Sciences, Budapest, HUNGARY kemeny@sztaki.hu

[iii]) Faculty of Information Technology, Péter Pázmány Catholic University, Budapest, HUNGARY kek@sztaki.hu

[iv]) Profactor Produktionsforschungs GmbH, Steyr, AUSTRIA werner.reisner@profactor.at

[v]) AnaLogic Computers Ltd., Budapest, HUNGARY bartfai@analogic-computers.com

[vi]) Maschinenbau Heinrich Hajek GmbH & Co KG, Bregenz, AUSTRIA

Abstract: **A new approach is proposed for vision-based sensing and processing for process control and monitoring of automated processes. The proposed approach relies on a number of binary logical sensors defined over specific regions of interest in the viewed scene. On top of these elementary sensors, temporal and logical aggregation mechanisms realize hierarchies of compound logical functions, able to detect complex events. Finally, scenario verification mechanisms are employed to monitor the occurrence order and timing of expected and actual events. The proposed framework has been tested and validated in an application involving monitoring of automated processes, demonstrating that the proposed approach provides a promising concept of vision-based event detection. The described framework is being implemented on the Bi-i standalone cellular vision system which has the potential of replacing several conventional sensors used for process control and fault detection in automation.**

Key-Words: **high speed image processing and analysis, image understanding, image sensors, process automation and monitoring, vision systems.**

1. Introduction

Recent advances in computer vision techniques allow the extraction of high-level semantic information from video streams, contributing to improvement in a variety of applications including surveillance, vision-based human-computer interaction, etc. Event detection requires the interpretation of the "semantically meaningful object actions" [1]. This task can be accomplished if the gap between pure numeric features and the semantic level is bridged. Past work has mostly dealt with extracting object trajectories followed by supervised learning, applying parameterized models for actions [2,3], usually consisting of predefined dynamic patterns of movements learnt in an off-line training phase. However, as the nature of events varies depending on the application, event modeling becomes a very challenging task.

In this paper, a new approach to event detection and interpretation is proposed within a process monitoring framework. The approach has two significant advantages over past work, as it *i)* decouples the detection and the interpretation of events from explicit, computer-based detection and recognition, and *ii)* it depends on very simple, low-level vision processes which is a key to robust and efficient performance.

The proposed approach is based on the *Vision-Based Logical Sensors* (VBLSs) [4] which meet a binary decision of whether a specific property holds in a specific image region at a certain moment in time, such as "region illumination exceeds predefined threshold", "region changed with respect to the scene background", "region profile matches stored prototype", etc., corresponding to primitive events in a video stream. *Compound Logical Sensors* (CLSs) can then be built through *temporal and logical aggregation* applied to the values of VBLSs (or, recursively, other CLSs). *Temporal aggregation* creates a CLS by reasoning on the value of a VBLS (or a CLS) over time, while *logical aggregation* creates a CLS by combining the values of several other VBLSs or CLSs into Boolean expressions.

The framework proposed is particularly suited to the application area of monitoring of automated processes. In most such processes (e.g., in mass production), things occur in a relatively strict, predetermined, scheduled way in comparison to other real life cases (e.g., vehicle control on highways). This permits us to turn difficult detection problems into much simpler verification problems, i.e., instead of trying to detect "what is going on" in the viewed scene, the VBLS approach can be used to verify that things proceed as expected. This results in several advantages:

- *Computational efficiency:* The VBLS approach requires simple, low level, computationally cheap, data parallel image processing operations to be applied on (typically) small image regions.
- *Extensibility:* VBLSs and CLSs can be dynamically tailored and expanded based on the needs of different application domains.
- *Flexibility and adaptability:* Most complex vision algorithms either fail in specific settings or require elaborate, non-intuitive parameter tuning. With the VBLS approach, it is more likely to come up with a suitable arrangement of VBLSs/CLSs.

Answering the need for an easily programmable, flexible sensor system with a minimum of required mechanical adjustment, the proposed solution involves the Bi-i standalone cellular vision system [5] which, enhanced with the VBLS framework, will be able to visually interpret image sequences of automated processes, report malfunctions and deviations in the process and assist the task of programming an adequate sequence of events. A significant advantage of this system is that it can potentially replace several "conventional" sensors and thus substantially reduce the set-up time and costs for assembly machines.

The paper is organized as follows. Section 2 describes briefly the issue of event detection in general. Section 3 provides a description of the basic elements of the VBLS approach as well as a search method to find correct sensor timing parameters in a semi-automatic way. Section 4 describes experiments carried out with a prototype implementation of the approach in the application area of the monitoring of automated processes. The document is concluded with a discussion summarizing the contributions of this work and future research directions.

2. Event detection

In process control, sensors are used whenever uncertainties are high enough to justify feedback. Traditionally, such processes are supervised using several simple sensors (e.g., light barriers), either to deliver actual values for closed control loops, or to detect fault situations which result in (discrete) decisions taken by the controller. The number and type of sensors employed depend on the given control task. In general, it can be said that there should be a bijection between the situations to be detected and the information the sensors provide. Typical situations are for example:

- Object was/was not inserted into the machine,
- Object got stuck or was dropped,
- Object was/was not gripped correctly,
- Object started/stopped moving, etc.

Some of these situations can be detected visually – this is the subset where "conventional" sensors can be directly replaced by vision. To accomplish this, relevant information must be extracted from an image series and categorized as an event, possibly related to initial object detection, such as:

- Object appeared (at a given position, with a given orientation and a given linear/angular velocity).
- Object started/is moving/rotating (at a given location, with a given linear/angular velocity).
- Object stopped (at a given location, with a given orientation).
- Object disappeared (moved out of the image or behind a non-transparent object in the background).
- Two (or more) objects move together or have a given relative linear/angular velocity.
- Objects joined/split up/are overlapping.

Finally, there may be other types of events related to the deformation of the shape of an object or combinations of any of the above (and other) events. This event list is the result of experiences collected through the study and analysis of various production systems [7,8,9,10,11,12].

Comparing the above event list and manufacturing situations where a sensor is employed, one can recognize that there is rarely a one-to-one mapping between them, but events can be very well recognized based on the above list. Consequently, a solution is needed to translate the above primitives into complex events to be recognized. This can be a logical model having events as inputs and manufacturing situations as outputs, to be built up typically by an expert having know-how in both process automation and machine vision.

3. The VBLS approach

Having described the task of the vision system, the specification of the elements of the solution will follow.

3.1 Vision-Based Logical Sensors (VBLSs)

A *Vision-Based Logical Sensor* (VBLS) is the basic entity in the VBLS approach, applying a set of user-defined *Image Processing and Analysis Algorithms* (IPAs) which detect a measurable property in a user-defined *Region Of Interest* (ROI) and delivers a Boolean output depending on whether or not the given property met a predefined requirement.

3.2 Region Of Interest (ROI)

A ROI is an arbitrarily shaped, user-defined region in an image. We denote a ROI R, with $R \equiv ROI(I,M,X,Y,W,H)$, meaning a region of interest in image I, having a mask image M with dimensions $W \times H$ (giving a Boolean specification whether a given pixel belongs to the ROI), located at image position (X,Y), see also Fig. 3.1.

3.3 Image Processing and Analysis Algorithms (IPAs)

Having defined a ROI, the next step is to define the algorithm(s) that will be applied on it. We distinguish four categories of IPAs.

- *Preprocessing IPAs,* processing a grayscale image to enhance/improve it (filtering operations as Gaussian smoothing, averaging, median filtering, histogram equalization, etc).
- *Analysis IPAs,* operating on grayscale images to produce a binary image in which pixels having a certain property are differentiated from the rest (e.g., change detection algorithms, methods verifying whether pixels have an expected value, etc).
- *Post-processing IPAs,* operating on binary images and producing another binary image with certain desired properties (e.g., morphological operators as erosion, dilation, etc).
- *Decision IPAs,* typically taking a binary image, and producing a decision on the value (true/false) of the Vision-Based Logical Sensor.

Thus, the output of an elementary logical sensor in the VBLS approach is the Boolean result of a collection of IPAs applied over a ROI. More formally, a VBLS L computes a binary function f implemented through a series of IPAs that are applied to a ROI R, or $L \equiv f(R)$.

3.3 Compound Logical Sensors (CLSs)

The Compound Logical Sensors (CLSs) are processing the output of VBLSs (and, possibly, other CLSs), implementing two aggregation mechanisms. In *temporal aggregation* (TA), an "observation window" is sliding over inputs measured at a series of discrete points in time (video frames, in the most straightforward approach), performing the operation $CLS_i \equiv TA(CLS_j, A_{\min}, A_{\max}, T_2, T_1)$

This means that a new CLS (CLS_i) is built through temporal aggregation (TA) of the values of CLS_j. Its output will be true at time t if CLS_j was true at least $A_{\min}$ and at most $A_{\max}$ times over the time interval $[t-T_1, t-T_2]$. CLS_i reporting the current value of CLS_j is then a special case of TA with $CLS_i \equiv TA(CLS_j,1,1,0,0)$. In *logical aggregation,* a CLS is built based on the logical combination of the results of other LSs

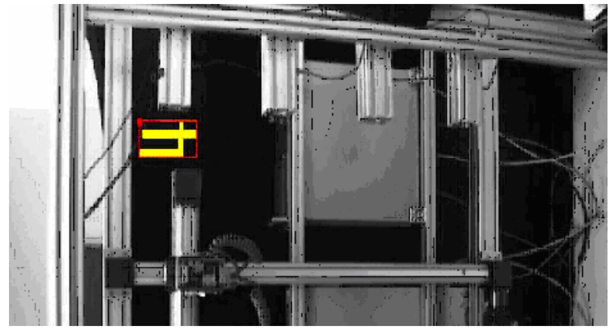

Fig. 3.1. Example of a ROI (red rectangle), with pixels selected for further processing in yellow

(VBLSs, or, recursively, CLSs). The following are some example CLSs:

$CLS_1 := LS_1$ OR LS_2

$CLS_2 := LS_3$ AND LS_4

$CLS_3 := LS_4$ XOR LS_5

$CLS_4 := CLS_1$ AND LS_6 (i.e., CLS_4 is equivalent to the expression "(LS_1 OR LS_2) AND LS_6").

Logical and temporal aggregation can be combined arbitrarily.

3.5 Scenarios (SC)

Scenarios are mechanisms provided to support the automatic monitoring of processes where several events occur serially, one after the other. A scenario SC is defined by the ordered list E of events $e_1,e_2,\ldots,e_n$ comprising it, the time differences d_i between the successive events e_i and e_{i+1}, and the time tolerances τ_i in the occurrence of these events. This means that if the event e_i occurs at time t_i, then, according to the scenario, the event e_{i+1} should occur in the time interval $[t_i+d_i-\tau_i, t_i+d_i+\tau_i]$. More formally, a scenario SC is represented by the triplet $SC \equiv (E,D,T)$, where $E = \langle e_1,e_2,\ldots,e_n \rangle$, $D = \langle d_1,d_2,\ldots,d_{n-1} \rangle$ and $T = \langle \tau_1,\tau_2,\ldots,\tau_{n-1} \rangle$. The validation of a scenario is achieved by a mechanism checking whether the events comprising the scenario have been detected and if their timing complies with the prescribed requirements. In the case of a *strict scenario*, the events comprising it should occur *only* with the predetermined timing, while in a *relaxed scenario*, they should occur *at least* at the required timing; however, some of the events could also occur at other time instances, besides the ones specified.

Regardless of its type, a scenario may fail either because an event was never detected, or because it was detected but did not occur at the proper timing. In both cases, the framework may provide an intuitive explanation for scenario failure, at different levels of detail. This is achieved by tracing the hierarchical structure

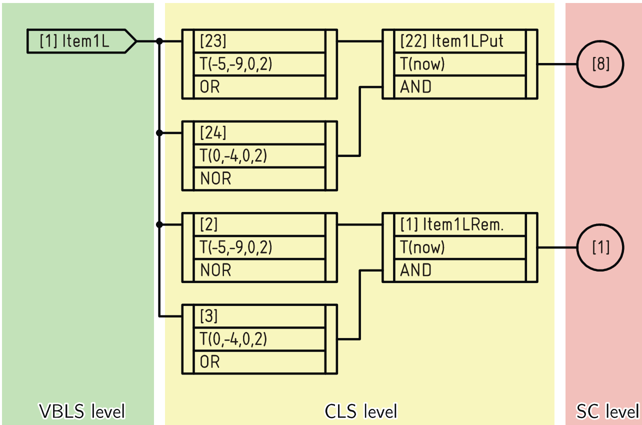

Fig. 3.2. Example detail of a VBLS—CLS—SC network with only one VBLS for input

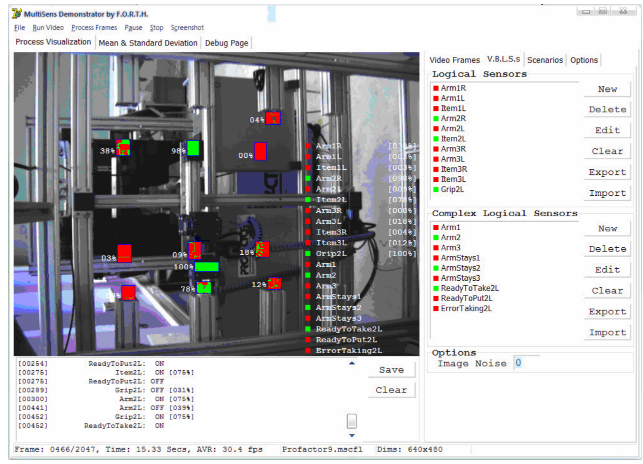

Fig. 3.3. Graphical interface of the software developed for testing the VBLS approach

of the CLS responsible for the non-detected event, and reporting the lower-level CLS or VBLS that did not produce the expected value.

3.6 Setting up the evaluation network

The above elements outline a three-level framework of logical evaluation, as shown in Fig. 3.2. While in most cases, a fair amount of experience, simple "rules of thumb" or brief practical tests are sufficient for setting correct IPA parameters, the same may not necessarily hold for the timing parameters in the TA operators and the events prescribed in the SC. Practical use would be largely facilitated by

- *a benchmarking method* expressing the qualities of the given timing configuration numerically, and
- *an optimization method* which can automatically find the timing parameters best fit for the given purpose (and using the aforementioned measure as an objective function).

Since fulfilling the latter requirement already gives a benchmarking tool as a by-product, the main goal pursued here was the elaboration of a suitable optimization method for finding suitable timing parameters. To this end, the timing properties of CLSs (both logical and temporal aggregation) were examined, resulting in linear inequalities which, together with basic assumptions about the finite length of a video sequence and causality in general, can define a *search space* where suitable timing parameters are to be found. Now, the optimal timing parameters must be determined for correct separation of successful input sequences from failures. Possible learning inputs may be

- one successful sequence;
- several successful sequences;
- several successful and unsuccessful sequences correctly labeled in advance by a human operator.

Having once specified the search space and possible inputs, a suitable objective function must be assembled which delivers its extremum when a given sequence of VBLS inputs fits best the requirements set by the timing parameters of the VBLS—CLS—SC network. The objective function is composed of the terms

- Q_1, or decision ability, expressing the ratio of correct pass/fail decisions for entire sequences;
- Q_2, or decision ability on the level of SC events, as a refinement of Q_1;
- Q_3, or decision quality, expressing how well the results are located within the given interval of correct decision;
- Q_4, or decision safety, expressing how far away a given result is located from the border of the corresponding binary decision.

From these, a compound criterion is assembled $Q=Q_1+Q_2+Q_3+Q_4$ where the above terms are scaled so that an optimization hierarchy is created with Q_1 receiving the highest priority and Q_4 the lowest one.

With the criterion, the search space and the learning patterns defined, the optimization procedure itself can be performed. Since the inequalities do not guarantee a convex search space and the possible choices of Q_3 and Q_4 include nonlinear functions, suitably robust optimization algorithms must be employed. In our case, the *Nelder-Mead simplex algorithm* [13] and the *Greedy tabu search method* [14] was tested with prerecorded learning sequences of all three possible compositions, the latter algorithm presenting better results. Experience obtained with the tests let us conclude that (semi-)automatic tools can be produced to assist the setup activity of installation personnel.

4. EXPERIMENTS

4.1 Validation of the VBLS approach

A software platform (Fig. 3.3) has been developed in order to test and validate the VBLS approach [4]. The capabilities of the platform include image sequence visualization, definition of ROIs, IPAs, VBLSs, and CLSs, as well as composition of scenarios (both strict and relaxed), parameter tuning, control of several visualization options, saving of results in text/video form and detailed error reporting. In addition, a framework for timing parameter search has been developed as a Matlab implementation with a separate interface specification.

The VBLS approach has been tested in the context of an application involving the monitoring of the activities of a 5-axis robot, consisting in moving a work piece from/to several locations.

A camera system was set up to observe and monitor the operation of the robot. Using the VBLS software platform, an unexperienced user is able to quickly define VBLSs, CLSs and scenarios to detect and verify several complex tasks of robot operation, including the correct recognition of failures (e.g., workpiece falling down or sticking to gripper or mounting frame, or even humans entering the field of view of the camera). Fig. 3.3 shows a typical screenshot of the system while in operation. LSs and CLSs which happen to be true/false at the particular moment in time are marked with green/red, respectively.

The fact that the recognition of complex events was already possible using some of the most simple IPAs (e.g., Gaussian smoothing) supports the conclusion that the power of the VBLS framework in detecting complex events lies in the spatial and temporal aggregation of the information of a large number of logical sensors and not in the “intelligence” of one, complex vision module – this property bearing a potential of efficient and robust performance in a wide variety of application domains.

4.2 The Bi-i Vision System and the VBLS approach

The Bi-i standalone cellular vision system [6] consists of state-of-the art cellular sensing, processing and communication devices enabling the system for application as a computing platform for combined topographic and non-topographic calculations in sensing—processing—actuation scenarios.

The Bi-i can capture and process up to 10,000 128×128 images per second. At its core is *Cellular Visual Technology* (CVT), which combines bio-inspired hardware and software solutions into a flexible and compact computational platform. Bi-i is ideally suited for applications requiring standalone operation and real-time performance in tasks whose complexity transcends the abilities of normal cameras and processors. Already proven in a variety of application pilots [5], the field of monitoring and control of automated processes constitutes a new application field where the Bi-i can unfold its ultra high-speed image processing capabilities.

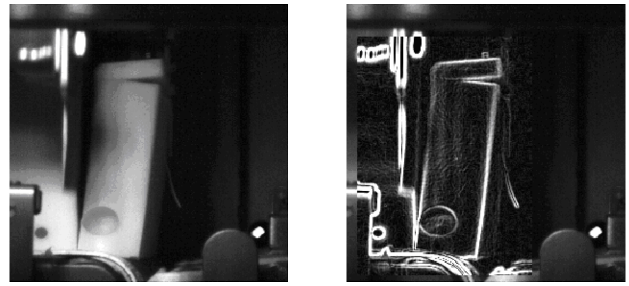

Fig. 4. 1. Image processing results obtained with the implementation of an IPA on Bi-i (right). The input image (left) is a frame of a video recording acquired in an industrial environment.

The Bi-i is well suited for high-speed event detection and process monitoring, since the current framework of the VBLS approach involves simple IPAs and the running time of them is in the range of a few dozens of microseconds per frame. Some IPAs are implemented (Fig. 4.1) in the platform independent Instant Vision library supporting application development for Bi-i, while further ones can easily be added to this software platform, working towards the final goal of integrating the smart-sensor-based vision system into a commercial product.

5. CONCLUSION

The paper presented a promising approach for replacing several sensors with a camera and a vision system in automated production lines. The approach takes advantage of the fact that in industrial production, numerous processes are clearly defined and their video images exhibit regular structures which can be captured by a series of simple filters (Vision-Based Logical Sensors, VBLSs), whose Boolean output is then passed on for interpretation by a network of Compound Logical Sensors (CLSs) employing both temporal and logical aggregation, and finally a Scenario (SC) whose elements prescribe the occurrence of discrete events. The work presented in the paper resulted in a prototype software which can be applied to build VBLS—CLS—SC networks and evaluate video sequences with them. In addition, a numerical tool was developed which supports the evaluation of the correctness of a given set of CLS and SC timing parameters, and gives semi-automatic assistance for finding best parameters with robust optimization algorithms. Hardware implementation of the Image Processing Algorithms (IPAs) comprising VBLSs was carried out on the Bi-i cellular image processing camera with binary VBLS outputs leaving the device instead of raw image data. The tests have successfully demonstrated the feasibility of the concept and may support plans of developing the vision system into a commercial product.

Acknowledgment

The authors wish to thank Kostas Hatzopoulos for his contribution in running most of the experiments, and Prof. Tamás Roska for his support, as well as the entire research and development team at AnaLogic Computers Kft. for the ongoing work related to the Bi-i system [6]. The work presented here was supported in part by the European 6th Framework Project MultiSens (contr. No. COOP-CT-2004-512668) as well as the Hungarian National Research Fund under grant No. OTKA T043547.

References

[1] F. Porikli, "Trajectory Distance Metric using Hidden Markov Model based Representation", in *Proc. of the PETS-ECCV'04 workshop, Prague, Czech Republic*, May 2004.

[2] G. Medioni, I. Cohen, F. Bremond, S. Hongeng, R. Nevatia, "Event Detection and Analysis from Video Streams", in *IEEE Trans. on PAMI*, Vol. 23 No. 8, Aug. 2001, pp. 873–889.

[3] S.O. Orphanoudakis, A.A. Argyros, M. Vincze "Towards a Cognitive Vision Methodology: Understanding and Interpreting Activities of Experts", in *ERCIM News, Special Issue on "Cognitive Systems"*, No 53, April 2003.

[4] T. Sarmis, A.A. Argyros, M.I.A. Lourakis, K. Hatzopoulos, "Robust and Efficient Event Detection for the Monitoring of Automated Processes", invited contribution, *Proc. of the VIE'2006 conference*, Bangalore, India, 26–28 September 2006.

[5] Á. Zarándy and Cs. Rekeczky, "Bi-i: A standalone ultra high speed cellular vision system", in *IEEE Circuits and Systems Magazine*, Vol. 5, No. 2, pp. 36–45, 2005.

[6] AnaLogic Computers Kft., Official web page. Available: http://www.analogic-computers.com

[7] Zs. J. Viharos, L. Monostori, K. Novák, G. Tóth, Z. Csongrádi, T. Kenderesy, T. Sólymosi, Á. Lőrincz, T. Koródi, "Monitoring of complex production systems, in view of digital factories", in *Proc. of the XVII IMEKO World Congress – Metrology in the 3rd Millennium*, pp. 1463–1468, 2003.

[8] L. Monostori, Zs. J. Viharos, "Hybrid, AI- and simulation-supported optimisation of process chains and production plants", in *Annals of the CIRP*, Vol. 50, No. 1, pp. 353–356, 2001.

[9] L. Monostori, Zs. J. Viharos, S. Markos, "Satisfying various requirements in different levels and stages of machining using one general ANN-based process model", in *Journal of Materials Processing Technology*, Vol. 107, No. 1–3, pp. 228–235, 2000.

[10] S. Markos, Zs. J. Viharos, L. Monostori, "Quality-oriented comprehensive modelling of machining processes", in *Proc. of the 6th ISMQC IMEKO Symposium on Metrology for Quality Control in Production*, pp. 67–74, 1998.

[11] L. Monostori, Zs. J. Viharos, "Quality-oriented modelling and optimisation of production processes and process chains", in *Proc. of the XVI. IMEKO World Congress*, pp. 439–444, 2000.

[12] Zs. J. Viharos, L. Monostori, T. Vincze, "Training and application of artificial neural networks with incomplete data", in *Proc. of the 15th International Conference on Industrial & Engineering Application of Artificial Intelligence & Expert Systems* (Book: *Lecture Notes of Artificial Intelligence*, Springer Computer Science Book, Springer-Verlag Heidelberg), pp. 649–659, 2002.

[13] J. Nelder and R. Mead, "A simplex method for function minimization", in *Computer Journal*, Vol. 7, pp. 308–311, 1965.

[14] F. Glover, "Tabu search: A tutorial", in *Interfaces, Special Issue on the Practice of Mathematical Programming*, Vol. 20, No. 1, pp. 74–94, 1990.