

Patterns in Temporal Series of Meteorological Variables Using SOM & TDIDT

Marisa Cogliati, Paola Britos and Ramón García-Martínez

Geography Department. School of Human Sciences. National University of Comahue
cogliati@uncoma.edu.ar

Software & Knowledge Engineering Center. Graduate School. Buenos Aires Institute of
Technology

Intelligent Systems Laboratory. School of Engineering. University of Buenos Aires
{pbritos,rgm}@itba.edu.ar

Abstract. The purpose of the present article is to investigate if there exist any such set of temporal stable patterns in temporal series of meteorological variables studying series of air temperature, wind speed and direction and atmospheric pressure in a period with meteorological conditions involving nocturnal inversion of air temperature in Allen, Río Negro, Argentina. Our conjecture is that there exist independent stable temporal activities, the mixture of which give rise to the weather variables; and these stable activities could be extracted by Self Organized Maps plus Top Down Induction Decision Trees analysis of the data arising from the weather patterns, viewing them as temporal signals.

1. Introduction

Classical laws of fluid motion govern the states of the atmosphere. Atmospheric states exhibit a great deal of correlations at various spatial and temporal scale. Diagnostic of such states attempt to capture the dynamics of various atmospheric variables (like temperature and pressure) and how physical processes influence the behaviour. Thus weather system can be thought as a complex system whose components interact in various spatial and temporal scales. It is also known that the atmospheric system is chaotic and there are limits to the predictability of its future state [Lorenz, 1963, 1965]. Nevertheless, even though daily weather may, under certain conditions, exhibit symptoms of chaos, long-term climatic trends are still meaningful and their study can provide significant information about climate changes. Statistical approaches to weather and climate prediction have a long and distinguished history that predates modelling based on physics and dynamics [Wilks, 1995; Santhanam and Patra, 2001]. Intelligent systems are appearing as useful alternatives to traditional statistical modelling techniques in many scientific disciplines [Hertz *et al.*, 1991; Rich & Knight, 1991; Setiono & Liu, 1996; Yao & Liu, 1998; Dow & Sietsma, 1991; Gallant, 1993; Back *et al.*, 1998; García Martínez & Borrajo, 2000; Grosser *et al.*, 2005]. In their overview of applications of neural networks (as example of intelligent system) in the atmospheric sciences, Gardner and Dorling [1998] concluded that neural networks

generally give as good or better results than linear methods. So far, little attention has been paid to combining linear methods with neural networks or other types of intelligent systems in order to enhance the power of the later. A general rule in this sort of applications says that the phenomenon to be learned by the intelligent system should be as simple as possible and all advance information should be utilized by pre-processing [Haykin, 1994]. This trend continues today with newer approaches based on machine learning algorithms [Hsieh and Tang, 1998; Monahan, 2000].

The term intelligent data mining [Evangelos & Han, 1996; Michalski *et al.*, 1998], is the application of automatic learning methods [Michalski *et al.*, 1983; Holsheimer & Siebes, 1991] to the non-trivial process of extract and present/display implicit knowledge, previously unknown, potentially useful and humanly comprehensible, from large data sets, with object to predict of automated form tendencies and behaviours; and to describe of automated form models previously unknown, [Chen *et al.*, 1996; Mannila, 1997; Piatetski-Shapiro *et al.*, 1991; 1996; Perichinsky & García-Martínez, 2000; Perichinsky *et al.*, 2003] involve the use of machine learning techniques and tools.

2. Problem

The central problem in weather and climate modelling is to predict the future states of the atmospheric system. Since the weather data are generally voluminous, they can be mined for occurrence of particular patterns that distinguish specific weather phenomena. It is therefore possible to view the weather variables as sources of spatio-temporal signals. The information from these spatio-temporal signals can be extracted using data mining techniques. The variation in the weather variables can be viewed as a mixture of several independently occurring spatio-temporal signals with different strengths. Independent component analysis (ICA) has been widely studied in the domain of signal and image processing where each signal is viewed as a mixture of several independently occurring source signals. Under the assumption of non-Gaussian mixtures, it is possible to extract the independently occurring signals from the mixtures under certain well known constraints. Therefore, if the assumption of independent stable activity in the weather variables holds true then it is also possible to extract them using the same technique of ICA. One basic assumption of this approach is viewing the weather phenomenon as a mixture of a certain number of signals with independent stable activity. By ‘stable activity’, meaning spatiotemporal stability, i.e., the activities that do not change over time and are spatially independent. The observed weather phenomenon is only a mixture of these stable activities. The weather changes due to the changes in the mixing patterns of these stable activities over time. For linear mixtures, the change in the mixing coefficients gives rise to the changing nature of the global weather [Stone, Porrill, Buchel, and Friston, 1999; Hyvarinen, 2001].

The purpose of the present article is to investigate if there exist any such set of temporal stable patterns related to the observed weather phenomena. Our conjecture is that there exist independent stable temporal activities, the mixture of which give rise to the weather variables; and these stable activities could be extracted by neural

networks analysis of the data arising from the weather and climate patterns, viewing them as temporal signals.

3. Proposed Solution

The variables as presented in the paper could not be considered random ones because of presence of temporal cycles. In addition, a linear behaviour as result of mixture of latent variables could not be assumed [Hyvarinen *et al.*, 2001]. In order to establish if there exist any such set of temporal stable patterns related to observed weather or climate phenomena we select weather station data described in [Flores *et al.*, 1996]. The records of the observed weather temporal series [Ambroise *et al.*, 2000; Malmgren and Winter, 1999 ; Tian *et al.*, 1999] are clustered with SOM [Kohonen, 2001; Kasi, *et al.*, 2000; Tirri, 1991; Duller, 1998] and rules describing each obtained cluster were built applying TDIDT [Quinlan, 1993] to each cluster records. The described process is shown in figure 1.

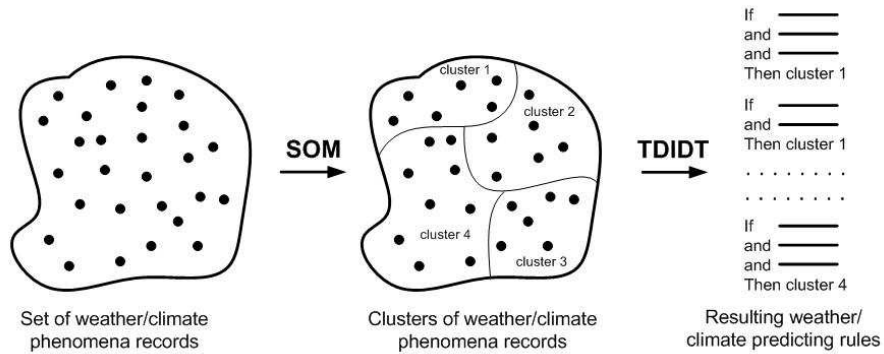


Fig. 1. Process for establishing temporal stable patterns related to observed weather/ climate phenomena

4. Data for experiments

The original data was a set of temperature, wind speed, wind direction and atmospheric pressure observations, taken every fifteen minutes from 13/10/94 to 17/10/94 in Allen, Río Negro province, Argentina. The weather station was located in the agricultural region called Upper Río Negro Valley (URNV) encompassing the lower valleys of the Limay and Neuquén rivers and the upper valley of the Negro river. The arable lands of best quality are located on the river terraces extending from the side pediments up to the floodplain. The terraces are limited by cliffs and the side pediments of the Patagonian plateau that surrounds the valleys. The valley is broad and shallow with steplike edges. The Negro river valley has a WNW-to-ESE orientation in the study area. The mean height differences with the North Patagonian Plateau is 120m for the Río Negro valley. The weather station data was obtained during MECIN (stands in spanish for: MEDiciones de la Capa de Inversión Nocturna:

Nocturnal Inversion Layer Measurements] field experience carried out in the URNV [Flores *et al.*, 1996] from September through October of the years 1992 to 1997. The data was complete, without using any replacement technique. The so called Upper Río Negro Valley in Argentina is one of the most important fruit and vegetable production regions of the country. It comprises the lower valleys of the Limay and Neuquén rivers and the upper Negro river valley. Out of the 41,671 cultivated hectares, 84.6% are cultivated with fruit trees, especially apple, pear and stone fruit trees [Cogliati, 2001]. Late frosts occurring when trees are sensitive to low temperatures have a significant impact on the regional production. This study presents an analysis of meteorological variables in one weather station in the Upper Río Negro Valley. To such effect, observations made when synoptic-scale weather patterns were favourable for radiative frosts (light wind and clear sky) or nocturnal temperature inversion in the lower layer were used. Calm winds were more frequent within the valleys. In Allen, air flow behaviour might be associated with forced channelling with wind direction following valley direction. In the night time, some cases of very light NNE wind occurred, which may be associated with drainage winds from the *barda*.

5. Results of experiments

The first analysis implementing SOM analysis determined nine clusters, that could be associated to different wind directions, maximum and mean wind speed, atmospheric pressure and temperature. Air temperature includes periodic daily variation, that was included in the analysis to explore relationship with wind variations. Four of nine groups identified, included the 94 percent of cases and several statistically significant rules. The detected rules for each group (cluster) are described in tables 1 to 9.

Groups A and B describe strongest wind cases with maximum wind speed greater than 5.8 m/s and mean wind speed greater than 1.3 m/s. Group C describe cases considering greater temperatures and wind speed with wind direction from south. Group D describes cases of wind speed up to 5 m/s from north to south directions and wind speed up to 5 m/s. In groups F and G and H cases present non obvious characteristics. Group J discriminates calm wind and groups Z_1 and Z_2 describes undetermined cases. The required frost analysis involve nocturnal and diurnal processes identification, so, the time of observation is a variable that might be included. The inclusion of date and time of observation produced a diminution of the quantity of groups involved, but an important increment in the number of rules (38 rules). This inclusion of new characteristics in the TDIDT analysis produced too much behaviour rules that produces confusion and detect obvious patterns as well as useful ones. This item would need further additional analysis. A confidence limit was pointed in order to study the rules.

Considering confidence level above 0.6 and rules involving more than 25 cases results in 11 rules. This rules pointed some groups characteristics. Group A includes 135 cases with relative higher air pressure mainly in the morning. The 324 cases in Group B present lower air pressure and wind speed. Prevailing wind direction was western sector. Group C discriminated weaker mean wind speed (less than 0.2 m/s) during the

morning and relative higher air pressure (371 cases) and cooler air temperature mainly from northern to southern direction. Group D presented westerly wind but cooler air temperature (96 cases) and F includes early afternoon and afternoon cases (154 cases).

RULES	SUPPORT DATA	CONFIDENCE
IF C5294P >= 992.84 AND C5294VVE < 0.20 AND HOUR < 10:33 THEN GROUP = A	26	0.85
IF C5294P >= 990.64 AND C5294P < 991.68 AND C5294VMX >= 0.65 AND HOUR >= 10:33 AND HOUR < 16:48 AND C5294TOU >= 16.75 THEN GROUP = A	12	0.92
IF C5294P >= 991.68 AND C5294VMX >= 0.65 AND HOUR >= 10:33 AND (1) (1) HOUR < 16:48 THEN GROUP = A	51	1
IF C5294P >= 989.61 AND C5294VVE >= 0.20 AND HOUR >= 6:57 AND HOUR < 10:33 THEN GROUP = A	26	1

Table 1. Rules from Group A (Cluster A)

RULES	SUPPORT DATA	CONFIDENCE
IF C5294P < 986.54 AND C5294VVE >= 0.20 AND HOUR < 10:33 THEN GROUP = B	44	0.91
IF C5294P < 989.24 AND C5294VMX >= 0.65 AND HOUR >= 10:33 AND C5294TOU >= 15.15 THEN GROUP = B	265	0.9
IF C5294P < 985.14 AND C5294VMX >= 0.65 AND C5294VMX < 1.55 AND HOUR >= 10:33 AND C5294TOU < 15.15 THEN GROUP = B	6	0.67
IF C5294P < 986.14 AND C5294VMX >= 1.55 AND HOUR >= 10:33 AND C5294TOU < 15.15 THEN GROUP = B	18	1
IF C5294P >= 986.14 AND C5294P < 989.24 AND C5294VMX >= 1.55 AND C5294VVE >= 1.10 AND HOUR >= 10:33 AND C5294TOU < 15.15 THEN GROUP = B	7	1
IF C5294P < 979.48 AND C5294VMX >= 0.65 AND C5294VVE < 0.20 AND HOUR < 8:09 AND C5294TOU < 10.35 THEN GROUP = B	2	1

Table 2. Rules from Group B (Cluster B)

RULES	SUPPORT DATA	CONFIDENCE
IF C5294P < 992.84 AND C5294VMX < 0.65 AND C5294VVE < 0.20 AND HOUR < 8:09 THEN GROUP = C	225	0.99
IF C5294P < 992.84 AND C5294VMX < 0.65 AND C5294VVE < 0.20 AND HOUR >= 8:09 AND HOUR < 10:33 AND C5294TOU < 5.40 THEN (1.00) (1) GROUP = C	5	1
IF C5294P >= 987.15 AND C5294P < 992.84 AND C5294VMX < 0.65 AND C5294VVE < 0.20 AND HOUR >= 8:09 AND HOUR < 10:33 AND C5294TOU >= 5.40 THEN GROUP = C	6	1
IF C5294P >= 984.49 AND C5294P < 987.15 AND C5294VMX < 0.65 AND C5294VVE < 0.20 AND HOUR >= 8:09 AND HOUR < 8:34 AND C5294TOU >= 5.40 THEN GROUP = C	4	0.75
IF C5294P >= 988.14 AND C5294P < 992.84 AND C5294VMX >= 0.65 AND C5294VVE < 0.20 AND (1) (1) HOUR >= 8:09 AND (1) (1) HOUR < 10:33 THEN GROUP = C	29	0.83
IF C5294P < 992.84 AND C5294VMX >= 0.65 AND C5294VVE < 0.20 AND HOUR < 8:09 AND C5294TOU >= 10.35 THEN GROUP = C	50	0.74
IF C5294P >= 989.61 AND C5294VVE >= 0.20 AND HOUR < 6:57 THEN GROUP = C	35	0.60
IF C5294P >= 989.05 AND C5294P < 992.84 AND C5294VMX >= 0.65 AND C5294VVE < 0.20 AND HOUR < 8:09 AND C5294TOU < 10.35 THEN GROUP = C	15	1
IF C5294P >= 989.24 AND C5294P < 990.64 AND C5294VMX >= 0.65 AND C5294VVE < 0.20 AND HOUR >= 10:33 AND HOUR < 16:48 THEN GROUP = C	12	0.92

Table 3. Rules from Group C (Cluster C)

RULES	SUPPORT DATA	CONFIDENCE
IF C5294P >= 979.48 AND C5294P < 989.05 AND C5294VMX >= 0.65 AND C5294VVE < 0.20 AND HOUR < 8:09 AND C5294TOU < 10.35 THEN GROUP = D	25	0.68
IF C5294P >= 986.54 AND C5294P < 989.61 AND C5294VVE >= 0.20 AND HOUR < 10:33 THEN GROUP = D	8	0.63
IF C5294P >= 989.24 AND C5294VMX >= 2.00 AND HOUR >= 16:48 THEN GROUP = D	24	0.67
IF C5294P >= 989.24 AND C5294P < 990.64 AND C5294VMX >= 0.65 AND C5294VVE >= 0.20 AND HOUR >= 10:33 AND HOUR < 16:48 THEN GROUP = D	6	0.67
IF C5294P >= 990.64 AND C5294P < 991.68 AND C5294VMX >= 0.65 AND HOUR >= 10:33 AND HOUR < 16:48 AND C5294TOU < 16.75 THEN GROUP = D	8	0.63
IF C5294P >= 986.14 AND C5294P < 989.24 AND C5294VMX >= 1.55 AND C5294VVE < 1.10 AND HOUR >= 10:33 AND C5294TOU < 15.15 THEN GROUP = D	26	0.81
IF C5294P >= 984.92 AND C5294P < 988.14 AND C5294VMX >= 0.65 AND C5294VVE < 0.20 AND HOUR >= 8:09 AND HOUR < 10:33 THEN GROUP = D	7	1

Table 4. Rules from Group D (Cluster D)

RULES	SUPPORT DATA	CONFIDENCE
IF C5294P >= 981.17 AND C5294VMX < 0.65 AND HOUR >= 12:45 THEN GROUP = F	159	1
IF C5294P >= 985.14 AND C5294P < 989.24 AND C5294VMX >= 0.65 AND C5294VMX < 1.55 AND HOUR >= 20:24 AND C5294TOU >= 13.20 AND C5294TOU < 15.15 THEN GROUP = F	2	1

Table 5. Rules from Group F (Cluster F)

RULES	SUPPORT DATA	CONFIDENCE
IF C5294P < 981.17 AND C5294VMX < 0.65 AND HOUR >= 10:33 THEN GROUP = G	15	0.93
IF C5294P >= 985.14 AND C5294P < 989.24 AND C5294VMX >= 0.65 AND C5294VMX < 1.55 AND HOUR >= 20:24 AND C5294TOU < 13.20 THEN GROUP = G	12	0.92
IF C5294P >= 989.24 AND C5294VMX >= 0.65 AND C5294VMX < 2.00 AND HOUR >= 16:48 THEN GROUP = G	20	0.4

Table 6. Rules from Group G (Cluster G)

RULES	SUPPORT DATA	CONFIDENCE
IF C5294P >= 981.17 AND C5294VMX < 0.20 AND HOUR >= 10:33 AND HOUR < 12:43 THEN GROUP = H	10	0.9
IF C5294P < 984.49 AND C5294VMX < 0.65 AND C5294VVE < 0.20 AND HOUR >= 8.09 AND HOUR < 10:33 AND >= 5.40 THEN GROUP = H	11	1
IF C5294P >= 984.49 AND C5294P < 992.84 AND C5294VMX < 0.65 AND C5294VVE < 0.20 AND HOUR >= 9:36 AND HOUR < 10:33 AND C5294TOU >= 5.40 THEN GROUP = H	7	1
IF C5294P >= 984.49 AND C5294P < 987.15 AND C5294VMX < 0.65 AND C5294VVE < 0.20 AND HOUR >= 8:38 AND HOUR < 9:36 AND C5294TOU >= 5.40 THEN GROUP = H	5	1

Table 7. Rules from Group H (Cluster H)

RULES	SUPPORT DATA	CONFIDENCE
IF C5294P >= 985.14 AND C5294P < 989.24 AND C5294VMX >= 0.65 AND C5294VMX < 1.55 AND HOUR >= 10:33 AND HOUR < 20:24 AND C5294TOU < 15.15 THEN GROUP = J	4	0.5

Table 8. Rules from Group J (Cluster J)

RULES	SUPORT DATA	CONFI-DENCE	RULES	SUPORT DATA	CONFI-DENCE
IF C5294P >= 981.17 AND C5294VMX >= 0.20 AND C5294VMX < 0.65 AND HOUR >= 10:33 AND HOUR < 12:43 THEN GROUP = UNDETERMINATE	6	0.33	IF C5294P < 984.92 AND C5294VMX >= 0.65 AND C5294VVE < 0.20 AND HOUR >= 8:09 AND HOUR < 10:33 THEN GROUP = UNDETERMINATE	10	0.4

Table 9. Rules from Group Z₁ and Z₂ (indetermined cluster)

In “C5294...”, “C52” is the meteorological station code and “94” is the year (1994). In “C5294vdd”, “vdd” is the wind orientation. In “C5294vve”, “vve” is average wind intensity. In “C5294tou”, “tou” is air temperature (C°). In “C5294P”, “P” is pressure (hPa).

The figure 2 presents the maximum wind speed versus local time for the different groups selected. The discrimination of different meteorological situations could differentiate physical relationships in the analyzed cases, further analysis considering atmospheric temporal variations could improve the selection, discarding the obvious deterministic patterns.

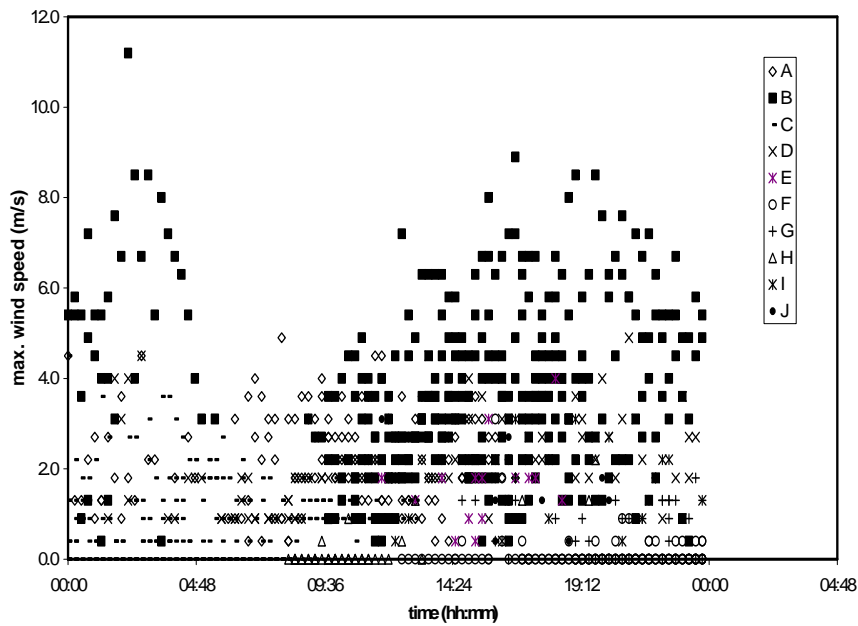


Fig. 2. Scatter plot of different groups data of maximum wind speed versus time in Allen (Río Negro Argentina) from 13/10/94 to 17/10/94.

6. Conclusions

The so called Upper Río Negro Valley in Argentina is one of the most important fruit and vegetable production regions of the country. It comprises the lower valleys of the

Limay and Neuquén rivers and the upper Negro river valley. Late frosts occurring when trees are sensitive to low temperatures have a significant impact on the regional production. Time series analysis of air temperature, atmospheric pressure, wind speed and direction involves a large amount of data and data mining could be an alternative to statistical traditional methods to find clusters with stable signals.

This study presents an analysis of meteorological variables in one weather station in the Upper Río Negro Valley by means of SOM analysis and applying TDIDT to build rules. To such effect, observations made when synoptic-scale weather patterns were favourable for radiative frosts (light wind and clear sky) or nocturnal temperature inversion in the lower layer were used. The obtained rules represent wind, temperature and pressure characteristics, the groups separate calm, and nocturnal and diurnal main characteristics according to prior traditional methods analysis (Cogliati, 2001), newer found relationships might be studied in advance.

The inclusion of a larger number of variables such time and date produces a large number of rules without defining precise intervals that produces confusion and detect obvious patterns as well as useful ones. This item would need further extensive study. The variation in the weather variables can be viewed as a mixture of several independently occurring spatio-temporal signals with different strengths.

Acknowledgements: The authors would like to thank Jorge Lässig for providing the meteorological data obtained in MECIN field experiment.

7. References

- Ambrose, C., Seze, G., Badran, F., and Thiria, S. 2000. *Hierarchical clustering of self-organizing maps for cloud classification*. Neurocomputing, 30(1):47–52
- Back, B., Sere, K., & Vanharanta, H. 1998. *Managing complexity in large data bases using self-organizing maps*. Accounting Management & Information Technologies 8, 191-210.
- Chen, M., Han, J., Yu, P. 1996. *Data mining: An overview from database perspective*. IEEE Transactions on Knowledge and Data Engineering 8(6): 866-883.
- Cogliati, M.G. 2001. *Estudio térmico y del flujo del aire en septiembre y octubre en los valles de los ríos Limay, Neuquén y Negro*. Doctoral Dissertation. University of Buenos Aires.
- Dow R. J. y Sietsma J. 1991. *Creating Artificial Neural Networks that Generalize*. Neural Networks . 4(1): 198-209.
- Duller, A. W. G. 1998. *Self-organizing neural networks: their application to "real-world" problems*. Australian Journal of Intelligent Information Processing Systems, 5:175–80
- Evangelos, S., Han, J. 1996. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. Portland, EE.UU.
- Flores, A. ; Lässig, J. ; Cogliati, M. ; Palese, C., Bastanski, M. 1996. *Mediciones de la Capa de Inversión Nocturna en los valles de los ríos Limay, Neuquén y Negro*. Proceedings VII Argentine Congress on Meteorology. VII Latinamerican and Iberic Congress on Meteorology. Buenos Aires.
- Gallant, S. 1993. *Neural Network Learning & Experts Systems*. MIT Press, Cambridge, MA.
- García Martínez, R. y Borrajo, D. 2000. *An Integrated Approach of Learning, Planning & Executing*. Journal of Intelligent & Robotic Systems. 29(1): 47-78.
- Gardner, M., Dorling, S. 1998. *Artificial neural networks (the multilayer perceptron) – a review of applications in the atmospheric sciences*. Atmospheric Environment 32: 2627-2636

- Grosser, H., Britos, P. y García-Martínez, R. 2005. *Detecting Fraud in Mobile Telephony Using Neural Networks*. Lecture Notes in Artificial Intelligence 3533: 613-615.
- Haykin, S., 1994. *Neural networks: A comprehensive foundation*. Prentice-Hall, Englewood Cliffs, NJ.
- Hertz J., A. Krogh y R. Palmer 1991. *Introduction to the Theory of Neural Computation*. Reading, MA: Addison-Wesley.
- Holsheimer, M., Siebes, A. 1991. *Data Mining: The Search for Knowledge in Databases*. Report CS-R9406, ISSN 0169-118X, Amsterdam, The Netherlands.
- Hsieh, W. , and Tang, B. 1998. *Applying neural network models to prediction and data analysis in meteorology and oceanography*. Bulletin of American Meteorological Society 79: 1855-1870.
- Hyvarinen, A. 2001. *Complexity pursuit: Separating interesting components from time-series*. Neural Computation 13: 883-898.
- Hyvarinen, A., Karhunen, J. and Oja, E. 2001. *Independent Component Analysis*. John Wiley & Sons.
- Kaski, S., Venna, J., and Kohonen, T. 2000. *Coloring that reveals cluster structures in multivariate data*. Australian Journal of Intelligent Information Processing Systems, 6:82-8.
- Kohonen, T. 2001. *Self-Organizing Maps*. Springer Series in Information Sciences, Vol. 30, Springer, Berlin.
- Lorenz, E. 1963. *Deterministic non-periodic flow*. Journal of Atmospheric Sciences 20: 130-141.
- Malmgren, B. A. and Winter, A. 1999. *Climate zonation in Puerto Rico based on principal components analysis and an artificial neural network*. Journal of Climate, 12:977-85
- Mannila, H. 1997. *Methods and problems in data mining*. In Proc. of International Conference on Database Theory, Delphi, Greece.
- Michalski, R., Carbonell, J., Mitchell, T. 1983. *Machine learning I: An AI Approach*. Morgan Kaufmann, Los Altos, CA.
- Michalski, R.S., Bratko, I., Kubat, M. 1998. *Machine Learning and Data Mining, Methods and Applications*. John Wiley & Sons Ltd, West Sussex, England.
- Monahan, A. 2000. *Nonlinear principal component analysis by neural networks: Theory and applications to the Lorenz system*. Journal of Climate 13: 821-835.
- Perichinsky, G., García-Martínez, R. 2000. *A Data Mining Approach to Computational Taxonomy*. Proceedings Argentine Computer Science Researchers Workshp: 107-110.
- Perichinsky, G., Servetto, A., García-Martínez, R., Orellana, R., Plastino, A. 2003. *Taxomic Evidence Applying Algorithms of Intelligent Data Mining Asteroid Families*. Proceedings de la International Conference on Computer Science, Software Engineering, Information Technology, e-Business & Applications 308-315.
- Piatetski-Shapiro, G., Frawley, W., Matheus, C. 1991. *Knowledge discovery in databases: an overview*. AAAI-MIT Press, Menlo Park, California.
- Piatetsky-Shapiro, G., Fayyad, U.M., Smyth, P. 1996. *From data mining to knowledge discovery*. AAAI Press/MIT Press, CA.
- Quinlan, R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers. San Mateo California.
- Rich E. y Knight, K. 1991. *Introduction to Artificial Networks*. Mac Graw-Hill. Publications.
- Santhanam M., and Patra, P. 2001. *Statistics of atmospheric correlations*. Physical Review E 64: 016102-1-1-7.
- Setiono R. & Liu. H. 1996. *Symbolic representation of neural networks*. IEEE Computer Magazine 29(3): 71-77.
- Stone, J., Porrill, J., Buchel, C., and Friston, K. 1999. *Spatial, temporal and spatiotemporal independent component analysis of fMRI data*. In 18th Leed Statistical Research Workshop on Spatiotemporal Mdeling and its Applications. University of Leeds.

- Tian, B., Shaikh, M. A., Azimi Sadjadi, M. R., Vonder Haar, T. H., and Reinke, D. L. 1999. *Study of cloud classification with neural networks using spectral and textural features*. IEEE Transactions on Neural Networks, 10(1):138–151
- Tirri, H. 1991. *Implementing Expert System Rule Conditions by Neural Networks*. New Generation Computing. 10(1): 55-71.
- Wilks, D. 1995. *Statistical methods in Atmospheric Sciences*. Academic Press, London.
- Yao X. y Liu Y. 1998. *Toward Designing Artificial Neural Networks by Evolution*. Applied Mathematics & Computation 91(1): 83-90.

Applying Genetic Algorithms to Convoy Scheduling

Edward M. Robinson¹ and Ernst L. Leiss²

1 Binary Consulting, Inc. 4405 East-West Highway, Suit 109,
Bethesda, MD 20814 USA, erobinson@binary-consulting.com

2 Ernst L. Leiss, Dept. of Computer Science, University of Houston,
coscel@cs.uh.edu

Abstract. We present the results of our work on applying genetic algorithms combined with a discrete event simulation to the problem of convoy scheduling. We show that this approach can automatically remove conflicts from a convoy schedule thereby providing to the human operator the ability to search for better solutions after an initial conflict free schedule is obtained. We demonstrate that it is feasible to find a conflict free schedule for realistic problems in a few minutes on a common workstation or laptop. The system is currently being integrated into a larger Transportation Information System that regulates highway movement for the military.

1 Introduction

The objective of this work is to automatically remove conflicts from a convoy schedule. The technique applied was to use a genetic algorithm approach combined with a simulation engine and real world data.

1.1 Convoys

Convoys are used to move equipment and people from one point to another [1], with equipment being trucks and vehicles that can perform the movement. A large military container ship can carry 800 containers and 1200 vehicles, which must be unloaded and moved quickly from the port to their final destination. This movement is managed by a Movement Control Team (MCT) that must organize and schedule the convoys and, at the same time, must integrate the convoy movement with the ongoing transportation of services within their area of responsibility. Data obtained from experienced transporters (human operators) report that 100 convoys per day are not uncommon. The challenge facing the MCT is to schedule the convoys and daily movements so that roads are uniformly used and, more importantly, that two or more

convoys do not run into conflict for a given resource. The term used by transporters for removing conflicts from a schedule is “deconfliction”.

An example of a conflict occurs when two convoys attempt to exit the same gate (assuming that only a single convoy at a time will use the gate) at the same time. Other conflicts include two convoys trying to cross (at right angles) through a single intersection, one convoy passing another on a single route leg, and two convoys merging onto the same highway segment. While these conflicts are most common, other issues can arise based on local rules and regulations so the system must be able to be extended to support these cases. Our system currently detects convoys attempting to merge or cross at a single node and one convoy overtaking another. Military doctrine restricts convoys to one at a time on a given road segment in most cases and further requires a 20-30 minute gap between convoys. This aspect is being added to the automatic conflict removal module as it is being integrated into the target system.

The workflow for convoy scheduling starts when a convoy commander submits a request for a convoy clearance to the MCT. This request includes a list of trucks and other vehicles, a route or strip map, the origin and destination, and a requested time of departure. The MCT will either grant the request without changes or change the departure time or route if the situation requires it.

1.2 Convoy Scheduling

To determine if convoys will run into conflicts, information is supplied regarding the speed of the convoys, the maximum speed on each road segment, the number of vehicles and their dimensions, the required gaps between vehicles, and the routes in a common format. The length of the convoy and speed along a given segment (the lesser of the convoy speed and segment speed) can be used along with the convoy length to calculate the pass time of the convoy (the time elapsed between lead vehicle and trail vehicle crossing the same point). Figure 1 illustrates convoy structure and length.

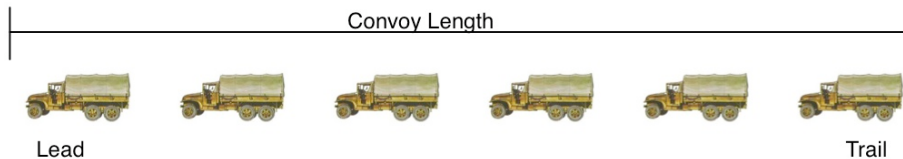


Fig. 1. Example of convoy organization.

Using the speed of the convoy and the pass time each convoy can be stepped forward in time along its given route tracking the times that the lead and the trail pass a given node. A conflict occurs when the lead from another vehicle reaches a node between the crossings of the lead and trail of the original convoy. Another conflict will occur if one convoy passes another on a single segment. This can be observed if the convoys reach the node at the end of the segment in a different order

than they passed the node at the beginning of a segment. We coin the term “inversion” for this kind of conflict.

1.3 Goals

Our primary goal for this project was to create a module for a Transportation Information System (TIS) that automatically adjusts an initial schedule to remove conflicts. Allowable changes include adjusting the departure time or selecting an alternate route (with the same origin and destination). The suitability of new schedules is ranked according to removal of all conflicts followed by a weighted evaluation of the changes to the schedule including closeness to requested departure time, convoy priority, and number of route changes.

Additional goals include support for incremental rescheduling based on changes in the field and extensibility to take into account local rules, guidelines, and opportunities (such as using roadside rest areas to allow one convoy to pass another). A final, but operationally vital constraint for all modules in the TIS is that the system must be field portable, i.e., it must be able to operate on a laptop without Internet access. Implied by this is a relative speedy operation; execution times of several hours are not conducive to responsiveness to developments and changes in the field. Execution times within 10 minutes are considered acceptable as conveyed by experienced transportation personnel.

2 Related Work

Convoy deconfliction is unique enough that there is limited amount of research on the subject. However, there do exist systems that automatically remove conflict from schedules as well as papers describing work on building conflict free convoy schedules while minimizing total time.

2.1 MOBICON

MOBICON [1,2] is a mature system used for scheduling convoys within the continental United States (CONUS). There are no publications that discuss the algorithm used to remove conflicts from convoy schedules. However, it is a reasonable assumption that MOBICON performs this function. MOBICON is tightly integrated with the rules and regulations for executing convoys and obtaining permission from each state to allow the convoy to use its highways; this would make it difficult to extract an extensible core algorithm. Also, MOBICON is a mainframe application, which clearly violates the requirement to be field portable.

2.2 Convoy Movement Problem (CMP)

The Convoy Movement Problem (CMP) attempts to find the minimum overall time for routing multiple convoys from the origins to destinations. [3] showed that

the problem is NP-complete in simple cases, and more complicated in more realistic situations. The fundamental culprit is the aimed-for optimality of a solution. Optimality is not required – at present, convoys are scheduled manually, with very limited computational support; therefore, the presently applied scheduling algorithms are obviously not optimal. Consequently, it is far more useful to apply heuristics in some form, which will improve solutions, but do not necessarily guarantee optimality. The approach taken in [4] recognized this and combined genetic algorithms with a branch and bound approach; however, the results, especially the time requirements of the programs rendered it rather impractical.

3 Our Approach

Our approach to meeting the objectives described in section 1.3 combines genetic algorithms with a simulation engine. The decision to use genetic algorithms (GA) was based on discussions with domain experts in convoys and interaction in the past with researchers in the GA area that emphasized the speed of recalculation in the face of changing conditions. The work done in [3, 4] supports our decision. Also, the notion of modifying a DNA string closely aligns with modifying a string of offset times. The results of the genetic algorithm directly communicated to the domain expert in the field with no translation other than appending the unit of time and name of convoy to the result.

There exist many conflict free schedules for a given set of convoys (a trivial approach would be to start each convoy on a different day). An optimal schedule was believed to be too expensive to calculate (since at best it is NP-complete) but discussions with domain experts determined that an optimal schedule was *not* the goal. The goal was to find a conflict free schedule that favored the priorities and original request times. Such a goal is ideal for a heuristic approach.

Further discussion with domain experts showed that the conditions for conflict and techniques for working around conflict change from location to location. The ideal system would support easy extension and the ability to adapt local business rules into the solver. Experience recommended the use of a discrete event simulation as the method for evaluating a schedule for conflicts. The simulation approach has the benefit of being easy to explain and validate with the domain experts as changes were made. The ability to refine the fitness function with data collected from probes inserted into the simulation to monitor key events and the ability to extend the simulation to handle new constraints and restrictions has been a major win over the more simplified closed forms of convoy interaction.

3.1 Genetic Algorithm Structure

The DNA string in our genetic algorithm consists of offset times to be applied to the requested departure time to determine the actual departure time for the convoy. This offset is taken from a set of offsets, which is a configuration parameter. A nice benefit of this approach is that the resulting DNA string when combined with the convoy names is straightforward to read (ex: convoy 1 starts 15 minutes earlier,

convoy 2 starts on time, convoy 3 starts 20 minutes later). In practice, the offsets are generally in multiples of 5, 10, or 15 minutes.

Mutation was applied by randomly selecting a given offset entry and selecting a randomly different offset time. For crossover, two “parents” are selected at random from the top half of the population and crossing the parent DNA strings at a randomly selected crossover point creates two new “children”. Each single string represented an alternate schedule and individual offsets (convoy) were treated independently.

Randomly selected offsets were used to create the initial population strings along with a single zeroed out string to represent the schedule “as-is” to ease tracking performance. The following steps were applied to each generation:

- Mutate
- Breed
- Evaluate
- Sort

The evaluation step was used to determine the fitness of each string (schedule), which combined a simple evaluation of weights with the simulation of the schedule to determine the number and type of conflicts.

The initial fitness function was determined through interviews with experienced transportation personnel to closely support transportation mission objectives. Inversions were considered worse than contention for a given node and the presence of any conflict was considered worse than other considerations such as closeness of actual departure time to requested departure time, closeness of predicted arrival time to requested arrival time, and favoring a higher priority to selected convoys that might carry fuel or ammunition. Several weights and formulas were tested for convergence on sample data with the final technique to calculate fitness being

$$\sum \frac{100}{inversions} + \frac{10}{conflicts} + \frac{1}{\sum offsets}$$

3.2 Simulation

A discrete event simulation (see Fig. 2) was used to step each convoy through its route starting at the time determined by adding the offset to its requested departure time. Active agents were used to model the lead and trail vehicles; route nodes and legs (edges) were modeled as resources, which were used to track which convoy was utilizing the node at a given time. A global simulation clock was used to determine which event occurs next and to issue an event trigger to the agent that had scheduled the event. Node resources allow locking by the head of a convoy and unlocking by the trail of a convoy. If the node is already locked (indicating that this resource is already being used by another convoy), the node resource will flag the conflict with a monitor that keeps count of all conflicts encountered.

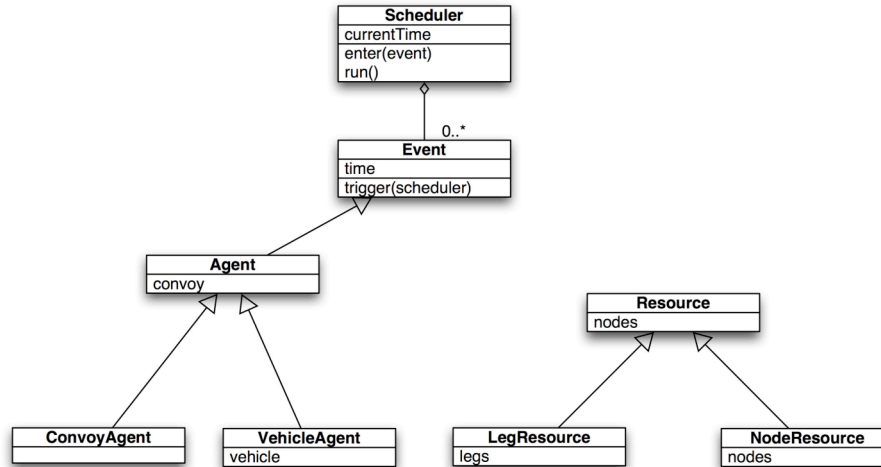


Fig. 2. Class model used to execute the simulation.

Simple queues located with leg resources are used to detect inversions. A convoy is added to the queue when it enters a leg of a route; it is removed from the queue when it exits it. A convoy that overtakes the first convoy would attempt to remove itself out of order and the leg resource would register that with a monitor that keeps a count of inversions.

3.3 Convoy Data

Convoy data were taken from the database used to test the TIS as well as generated from templates. Also, data collected from a simulation of a single convoy were vetted against documented training examples. These data were stored in an object model serving the simulation. The object model's class diagram is illustrated in Figure 3.

The convoy schedule holds all of the convoys and their information. Routes are shared amongst the convoys since that is the case in the field. Additionally, military doctrine allows for convoys to be hierarchically structured into march-units, serials, and columns. This is modeled with a tree structure in the object model to simplify logic. Vehicle information is used to provide the length of the vehicle, which is used in turn to calculate convoy length and pass time. Additionally, the height and weight of a vehicle is needed to determine the validity of switching routes. If an alternate route has a low or weak bridge the convoy may not be able to traverse that route. Also, cargo contents may impact the ability of a convoy to cross a given leg. For example, fuel trucks are not allowed to pass near water in some European countries.

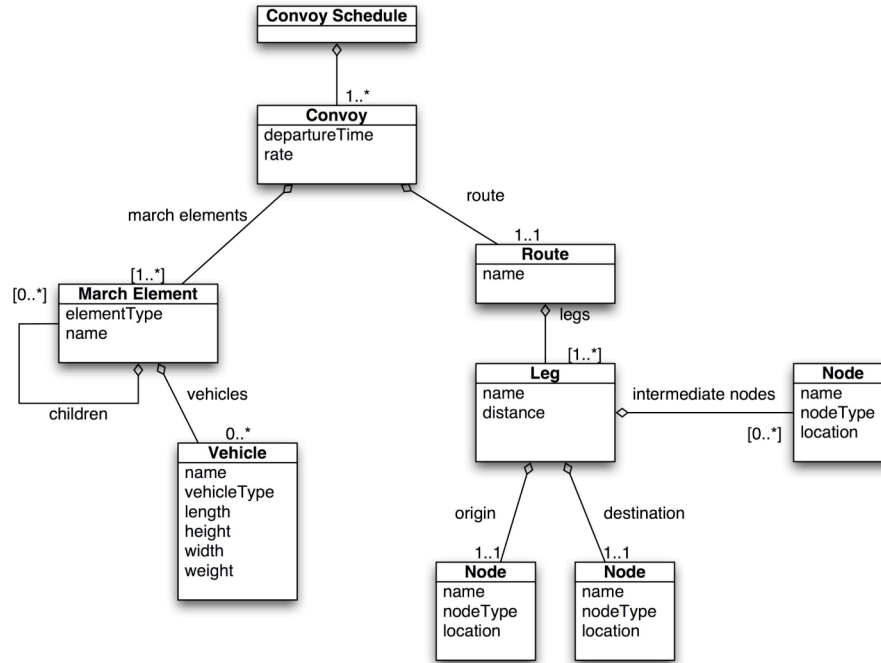


Fig. 3. Object model used for representing convoy in simulation.

4 Results

We tested our system using convoy sizes of 10, 25, and 50 each with 30 vehicles. We were able to reach conflict free schedule within 50 generations for each test case by varying the size of the time window. The results of these runs are given in Table 1. Each convoy requests departure between 9AM and 8PM on a single day. In each case, a conflict free schedule is found within a minute on a reasonably powerful workstation (a 2.7 GHz dual processor PowerPC). The code is written in Java 1.4.2 with no performance tuning or the use of the HotSpot server runtime. Tests were run on Pentium 4 laptops used in the field with comparable performance (within 2 minutes maximum processing time).

Table 1. Window sizes were adjusted to find a conflict free solution in fewer than 50 steps.

Number of convoys – Size	Steps to first conflict free	Time window in minutes	Average offset in minutes	Elapsed time in seconds
10 – 30	23	150	79	8.3
25 – 30	40	480	288	33.8
50 – 30	36	2160	1131	57.7

The time window, which is the maximum amount of time that a convoy schedule entry can be delayed or advanced, has a significant impact on the performance of the search algorithm. We constructed a test to compare the impact of window size on performance on the 25-convoy test data. The results in Table 2 show this performance. Each test run took approximately 2 minutes of user time to complete 150 steps.

Table 2. Test case showing the effect of changing window size on 25 convoys.

Time window in minutes	Steps to first conflict free	Average offset at step 50	Average offset at step 100	Average offset at step 150
360	62	141.3	109.6	110.0
390	36	170.2	121.7	92.9
420	31	178.2	135.6	98.0
450	44	233.1	154.6	117.1
480	46	173.0	150.0	125.7

Clearly the system is sensitive to the time window size with an apparent minimum average offset. This demonstrates the utility of having a human operator in the loop. The operator can guide the solution according to the tactical situation. If speed is the preeminent condition, the operator will stop the simulation as soon as a conflict free schedule is achieved. However, if more time is available, the operator can adjust the time window as well as allow the simulation to run longer in order to achieve a better solution.

5 Conclusion and Future Work

We have shown that a field portable system can be implemented to automatically find a conflict free schedule given an initial convoy schedule. The system is able to find a solution within a minute for up to 50 convoys and can continue searching according to the needs of and as directed by the human operator. This system is currently being integrated into an existing Transportation Information System in order to be fielded in the 2006 to 2007 timeframe.

In the future, we will address improving the performance of the code through standard tuning techniques and developing a larger set of test scenarios to analyze common situations (such as unloading large container ships or return trip deployment). We will also investigate different techniques to manage the sensitivity of the time window to finding better solutions. Finally, the transportation community is interested in utilizing the same techniques for non-convoy transportation uses such as supply chain management and backhaul optimization.

6 References

1. Army Field Manuals: FM 55-1, FM 55-30, and FM 55-65. GlobalSecurity.org
<http://www.globalsecurity.org/military/library/policy/army/fm/index.html>
2. R.J. Shun, Army Logistics Management College (ALMC). Automating Convoy Operations. <http://www.almc.army.mil/alog/issues/NovDec97/MS214.htm>
3. P. Chardaire, G.P. McKeown, S.A. Verity-Harrison, and S.B. Richardson, "Solving a Time-Space Formulation for the Convoy Movement Problem", *Operations Research*, vol. 53, no. 2, pp. 219-230, 2004.
4. Y.N. Lee, G.P. McKeown, and V.J. Rayward-Smith, The Convoy Movement Problem with Initial Delays, *Modern Heuristic Search Methods* (John Wiley & Sons, 1996).

A GRASP algorithm to solve the problem of dependent tasks scheduling in different machines

Manuel Tupia Anticona
Pontificia Universidad Católica del Perú
Facultad de Ciencias e Ingeniería, Departamento de Ingeniería, Sección
Ingeniería Informática
Av. Universitaria cuadra 18 S/N
Lima, Perú, Lima 32
tupia.mf@pucp.edu.pe

Abstract. Industrial planning has experienced notable advancements since its beginning by the middle of the 20th century. The importance of its application within the several industries where it is used has been demonstrated, regardless of the difficulty of the design of the exact algorithms that solve the variants. Heuristic methods have been applied for planning problems due to their high complexity; especially Artificial Intelligence when developing new strategies to solve one of the most important variants called task scheduling. It is possible to define task scheduling as: .a set of N production line tasks and M machines, which can execute those tasks, where the goal is to find an execution order that minimizes the accumulated execution time, known as makespan. This paper presents a GRASP meta heuristic strategy for the problem of scheduling dependent tasks in different machines

1 Introduction

The task-scheduling problem has its background in industrial planning [1] and in task-scheduling in the processors of the beginning of microelectronics [2]. That kind of problem can be defined, from the point of view of combinatory optimization [3], as follows:

Considering M machines (considered processors) and N tasks with T_{ij} time units of duration for each i -esim task executed in the j -esim machine, we wish to program the N tasks in the M machines, trying to obtain the most appropriate execution order,

fulfilling certain conditions that satisfy the optimality of the required solution for the problem.

The scheduling problem presents a series of variants depending on the nature and the behavior of both, tasks and machines. One of the most difficult to present variants, due to its high computational complexity is that in which the tasks are dependent and the machines are different.

In this variant each task has a list of predecessors and to be executed it must wait until such is completely processed. We must add to this situation the characteristic of heterogeneity of the machines: each task lasts different execution times in each machine. The objective will be to minimize the accumulated execution time of the machines, known as *makespan* [3]

Observing the state of the art of the problem we see that both its practical direct application on industry and its academic importance, being a NP-difficult problem, justifies the design of a heuristic algorithm that search for an optimal solution for the problem, since there are no exact methods to solve the problem. In many industries such as assembling, bottling, manufacture, etc., we see production lines where wait periods by task of the machines involved and saving the resource time are very important topics and require convenient planning.

From the previous definition we can present a mathematical model for the problem as in the following illustration and where it is true that:

- X_0 represents makespan
- X_{ij} will be 0 if the j -esim machine does not execute the i -esim task and 1 on the contrary.

Minimize X_0

$$\text{s.a} \quad X_0 \geq \sum_{i=1}^N T_{ij} * X_{ij} \quad \forall j \in 1..M$$

$$X_0 \geq \sum_{j=1}^M X_{ij} = 1 \quad \forall i \in 1..N$$

Fig. 1. A mathematical model for the task-scheduling problem.

1.1 Existing methods to solve the task-scheduling problem and its variants

The existing solutions that pretend to solve the problem, can be divided in two groups: exact methods and approximate methods.

Exact methods [6, 7, 8, 9] try to find a sole hierarchic plan by analyzing all possible task orders or processes involved in the production line (exhaustive

exploration). Nevertheless, a search and scheduling strategy that analyzes every possible combination is computationally expensive and it only works for some kinds (sizes) of instances.

Approximate methods [3, 4 and 5] on the other hand, do try to solve the most complex variants in which task and machine behavior intervenes as we previously mentioned. These methods do not analyze exhaustively every possible pattern combinations of the problem, but rather choose those that fulfill certain criteria. In the end, we obtain sufficiently good solutions for the instances to be solved, what justifies its use.

1.2 Heuristic Methods to Solve the Task-scheduling variant

According to the nature of the machines and tasks, the following subdivision previously presented may be done:

- Identical machines and independent tasks
- Identical machines and dependent tasks
- Different machines and independent tasks
- Different machines and dependent tasks: the most complex model to be studied in this document.

Some of the algorithms proposed are:

A Greedy algorithm for identical machines propose by Campello and Maculan [3]: the proposal of the authors is to define the problem as a discreet programming one (what is possible since it is of the NP-difficult class), as we saw before.

Using also, Greedy algorithms for different machines and independent tasks, like in the case of Tupia [10] The author presents the case of the different machines and independent tasks. Campello and Maculan's model was adapted, taking into consideration that there were different execution times for each machine: this is, the matrix concept that it is the time that the i -esim task takes to be executed by the j -esim machine appears.

A GRASP algorithm, as Tupia [11]. The author presents here, the case of the different machines and independent tasks. In this job the author extended the Greedy criteria of the previous algorithm applying the conventional phases of GRASP technique and improving in about 10% the results of the Greedy algorithm for instances of up to 12500 variables (250 tasks for 50 machines).

1.3 GRASP Algorithms

- GRASP algorithms (for *Greedy Randomized Adaptive Search Procedure*) are meta heuristic techniques. T. Feo and M. Resende developed such technique by the end of the 80's [5] While the Greedy criteria let us select only the best value of the objective function, GRASP algorithms *relax or increase* this criteria in

such a way that, instead of selecting a sole element, it forms a group of elements, that candidate to be part of the solution group and fulfill certain conditions, it is about this group that a random selection of some of its elements.

This is the general scheme for GRASP technique:

```

GRASP Procedure (Instance of the problem)
  1. While <stop-condition is not true> do
    1.1 Construction Phase ( $S_k$ )
    1.2 Improvement Phase ( $S_k$ )
  2. Return (Best  $S_k$ )
End GRASP

```

Fig. 2. General structure of GRASP algorithm

About this algorithm we can affirm:

Line 1: the GRASP procedure will continue while the stop condition is not fulfilled. The stop condition can be of several kinds: optimality (the result obtained presents a certain degree of approximation to the exact solution or it is optimal enough); number of executions carried out (number of interactions); processing time (the algorithm will be executed during a determined period of time).

Lines 1.1 and 1.2: the two main phases of a GRASP algorithm are executed, later: construction stage of the adapted random Greedy solution; and the stage of improvement of the previously constructed solution (combinatorial analyses of most cases).

2 Proposed GRASP Algorithm

We must start from the presumption that there is a complete job instance that includes what follows: a quantity of task and machines (N and M respectively); an execution time matrix T and the lists of predecessors for each task in case there were any.

2.1 Data structures used by the algorithm

Let us think that there is at least one task with predecessors that will become the initial one within the batch, as well as there are no circular references among predecessor tasks that impede their correct execution:

Processing Time Matrix T : (T_{ij}) $M \times N$, where each entry represents the time it takes the j -esim machine to execute the i -esim task.

Accumulated Processing Times Vector A: (A^i) , where each entry A^i is the accumulated working time of M_i machine.

P_k : Set of predecessor tasks of J_k task.

Vector U: (U_k) with the finalization time of each J_k task.

Vector V: (V_k) with the finalization time of each predecessor task of J_k , where it is true that $V_k = \max\{U_r\}, J_r \in P_k$

S_i : Set of tasks assigned to M_i machine.

E: Set of scheduled tasks

C: Set of candidate tasks to be scheduled

Fig. 3. Data structures used by the algorithm

We are going to propose to selection criteria during the development of the GRASP algorithm, what is going to lead us to generate two relaxation constants instead of only one:

- Random GRASP selection criteria for the best task to be programmed, using relaxation constant α .
- Random selection criteria for the best machine that will execute the task selected before, using an additional θ parameter.

2.2 Selection criteria for the *best task*

These criteria bases on the same principles as the Greedy algorithm presented before:

Identifying the tasks able to be programmed: this is, those that have not been programmed yet and its predecessors that have already been executed (or do not present predecessors).

For each one of the tasks able, we have to generate the same list as in the Greedy algorithm: accumulated execution times shall be established starting from the end of the last predecessor task executed.

The smallest element from each list must be found and stored in another list of local minimums. We shall select the maximum and minimum values of the variables out of this new list of local minimums: *worst and best* respectively.

We will form a list of candidate tasks RCL analyzing each entry of the list of local minimums: if the corresponding entry is within the interval $[\text{best}, \text{best} + \alpha * (\text{worst} - \text{best})]$ then it becomes part of the RCL. A task is chosen by chance out of those that form the RCL.

2.3 Selection criteria for the *best machine*

Once a task has been selected out of RCL, we look for the best machine that can execute it. This is the main novelty of the algorithm proposed. The steps to be followed are the next ones:

The accumulated time vector is formed once again from the operation of the predecessors of the *j*-esim task, which is being object of analyses. Maximum and minimum values of the variables are established: worst and best respectively.

Then we form the list of candidate machines MCL: each machine that executes the *j*-esim task in a time that is in the interval (best, best+ θ *(worst-best)) is part of the MCL. Likewise, we will select one of them by chance, which will be the executioner of the *j*-esim task.

2.4 Presentation of the algorithm

GRASP Algorithm_Construction (M, N, T, A, S, U, V, α , θ)

1. **Read and initialize N, M, α , θ , J_1, J_2, \dots, J_N , T, A, S, U, V**
2. **E = \emptyset**
3. **While $|E| \neq N$ do**
 - 3.1 **C = \emptyset**
 - 3.2 **best = $+\infty$**
 - 3.3 **worst = 0**
 - 3.4 **For l : 1 to N do**
 - If $(P_l \subseteq E) \wedge (J_l \in E) \Rightarrow C = C \cup \{J_l\}$
 - 3.5 **Bmin = \emptyset**
 - 3.6 **For each $J_t \in C$ do**
 - 3.6.1 **$V_L = \max_{J_l \in P_k} \{U_l\}$**
 - 3.6.2 **$B_{\min} = B_{\min} \cup \text{Min}_{p \in [1, M]} \{T_{pl} + \max\{A_p, V_l\}\}$**
 - End for 3.6**
 - 3.7 **best = $\text{Min}\{B_{\min}\}$ {Selection of the best task. Formation of RCL}**
 - 3.8 **worst = $\text{Max}\{B_{\min}\}$**
 - 3.9 **RCL = \emptyset**
 - 3.10 **For each $J_t \in C$ do**
 - If $\text{Min}_{p \in [1, M]} \{T_{pl} + \max\{A_p, V_l\}\} \in [\text{best}, \text{best} + \alpha^* (\text{worst} - \text{best})] \Rightarrow$
RCL = RCL $\cup \{J_t\}$
 - 3.11 **k = $\text{ArgRandom}_{J_l \in \text{RCL}}\{\text{RCL}\}$**
 - 3.12 **MCL = \emptyset {Selection of the best machine}**
 - 3.13 **best = $\text{Min}_{p \in [1, M]} \{T_{pk} + \max\{A_p, V_k\}\}$**
 - 3.14 **worst = $\text{Max}_{p \in [1, M]} \{T_{pk} + \max\{A_p, V_k\}\}$**
 - 3.15 **For i: 1 to M do**

If $T_{ik} + \max\{A_i, V_k\} \in [\text{best}, \text{best} + \theta^* (\text{worst-best})] \Rightarrow \text{MCL} = \text{MCL} \cup \{M_i\}$
3.16 $i = \text{ArgAleatorio}_{M_i \in \text{MCL}} \{\text{MCL}\}$
3.17 $S_i = S_i \cup \{J_k\}$
3.18 $E = E \cup \{J_k\}$
3.19 $A_i = T_{ik} + \max\{A_i, V_k\}$
3.20 $U_k = A_i$
End While 3
4. makespan = $\max_{k \rightarrow 1..M} A_k$
5. Return $S_i, \forall i \in [1, M]$
End GRASP Algorithm_construction

Comments on the GRASP Algorithm proposed:

- Line 1: entry of the variables e initialization of the data structures needed for the working instance as N, M, T, A, U, V, E, Si for each machine, α, θ etc.
- Line 2: the process ends when all the programmed tasks of the batch are in group E.
- Line 3.1: the list of apt tasks C is initialized empty.
- Line 3.2: best and worst variables that will be used to work the intervals of the relaxation criteria are initialized.
- Line 3.4: the list of apt tasks C is formed
- Line 3.5: the list of local minimums Bmin is initialized.
- Line 3.6 – 3.6.2: entries corresponding to list V are actualized; list Bmin is formed adding each minor element of the accumulated time list of each task.
- Lines 3.7 – 3.8: maximum and minimum values of Bmin are assigned to worst and best variables respectively.
- Line 3.9: RCL list is initialized empty
- Lines 3.10 – 3.11: RCL list is formed when the condition $\text{Min}_{p \in [1, M]} \{T_{pl} + \max\{A_p, V_l\}\} \in [\text{best}, \text{best} + \alpha^* (\text{worst-best})]$ is true, then an element from this list (k) is chosen by chance.
- Lines 3.12 – 3.16: minimum and maximum execution times for the task selected in 3.11 are chosen. MCL will form from the machines that execute such task fulfilling the condition: $T_{ik} + \max\{A_i, V_k\} \in [\text{best}, \text{best} + \theta^* (\text{worst-best})]$. In the end a machine is also chosen in an aleatoric way (variable i).
- Lines 3.17 – 3.20: structures E, A, U and S_i are actualized where it corresponds.
- Line 3: makespan is determined as the major entry of A.
- Line 4: assigned results found are given back.

3. Numeric Experiences

The instances with which the algorithm was tested are formed by an M quantity of machines, N of task and a T matrix of execution times. The values used for M and N , respectively, were:

- Number of tasks N : within the interval of 100 to 250, taking 100, 150, 200 and 250 values as points of reference.
- Number of machines M : a maximum of 50 machines taking 12, 25, 37 and 50 values as point of reference.
- Processing time matrix: it will be generated in a random way with values from 1 to 100 time units¹.

We have a total of 16 combinations for the machine-tasks combinations. Likewise, for each combination 10 different instances will be generated, which gives a total of 160 test problems solved.

3.1 Quality of the GRASP solution compared to a simply Greedy solution

As there is no literature on pre-determined test instances for such problem, we decided to confront the results with those of a Greedy algorithm². In the first summary table average results of Greedy and GRASP algorithms execution (in that order) over the respective instances are shown; CPU times consumed by both algorithms, the quantity of executed iterations in the GRASP Construction phase and the values assigned to constants α and θ ; finally the efficiency of the GRASP result over the Greedy result calculated as follows is also shown: $1 - (\text{GRASP Result} / \text{Greedy Result})$

Machine \ Task	Makespan Greedy	CPU Used Time	GRASP			Iterations (Average)	CPU Used Time	Efficient
			Makespan	α	θ			
100 \ 12	213.3	0.56	193.7	0.26	0.04	3750	102.53	9.189%
100 \ 25	113.7	0.59	108.6	0.24	0.01	3750	115.8	4.485%
100 \ 37	76.5	0.56	74.3	0.155	0.01	3750	127.6	2.876%
100 \ 50	59.1	0.52	57.8	0.145	0.01	3750	140.91	2.200%
150 \ 12	283.1	0.58	258.2	0.25	0.01	3000	151.18	8.795%
150 \ 25	125.2	0.52	114	0.21	0.01	3000	170.49	8.946%
150 \ 37	86.1	0.55	84.1	0.155	0.01	3000	201.7	2.323%
150 \ 50	68.6	0.53	67.4	0.115	0.01	3000	224.35	1.749%
200 \ 12	325.9	0.54	307	0.07	0.01	2500	216.83	5.799%
200 \ 25	127	0.57	117	0.247	0.01	2500	229.2	7.874%
200 \ 37	109.8	0.57	105.2	0.112	0.01	2500	248.75	4.189%
200 \ 50	76.4	0.47	74.6	0.155	0.01	2500	274.58	2.356%
250 \ 12	411.8	0.46	377.1	0.15	0.01	2500	249.2	8.426%
250 \ 25	183.5	0.58	168.2	0.28	0.01	2500	282.34	8.338%
250 \ 37	125.3	0.57	119.6	0.26	0.01	2500	309.03	4.549%
250 \ 50	96.5	0.55	91.1	0.123	0.01	2500	343.32	5.596%
Efficient								5.481%

Table 1. Summary table of GRASP vs. Greedy numeric experiments

¹ : We shall notice that a very high execution time (+ ∞) may be interpreted as if the machine does not execute a determined task. In this case an execution time equal to 0 may be confuse as the machine executes the task so fast that it does it instantaneously, without taking time

² : In order to have a Greedy behavior it is enough to make constant α equal to 0 without regardless of selection of the best machine; that is why the algorithm is not added.

3.2 Real quality of the GRASP solution confronted with mathematical model's solutions

In order to determine the real quality of the solutions we decided to apply the mathematical model of the problem to linear programming problem solver packages as LINDO tool in its student version, in order to obtain exact solutions and confront them with those of the proposed GRASP algorithm.

The two tables that follow summarize the exact results obtained confronted with the heuristic (voracious solution) and meta heuristic (GRASP solution) results. In addition to this we will present the values of the relaxation constants used for the GRASP construction phase where nearly 7000 iterations for all the cases were carried out.

Table 2. Experimental results for M = 3

Matrix	N	M	Lindo Model	Greedy Algorithm	GRASP Algorithm	% Lindo/Greedy	% Lindo/GRASP
6x3_0	6	3	28	35	35	25.00%	0.00%
6x3_1	6	3	67	67	67	0.00%	0.00%
6x3_2	6	3	75	87	85	16.00%	2.30%
8x3_0	8	3	40	45	44	12.50%	2.22%
8x3_1	8	3	88	102	102	15.91%	0.00%
8x3_2	8	3	64	82	64	28.13%	21.95%
12x3_0	12	3	130	162	162	24.62%	0.00%
12x3_1	12	3	121	134	121	10.74%	9.70%
12x3_2	12	3	80	108	89	35.00%	17.59%
15x3_0	15	3	162	201	168	24.07%	16.42%
15x3_1	15	3	125	137	125	9.60%	8.76%
15x3_2	15	3	164	190	176	15.85%	7.37%
Efficient mean						18.12%	7.19%

Table 3. Experimental results for M = 5

Matrix	N	M	Lindo Model	Greedy Algorithm	GRASP Algorithm	% Lindo/Greedy	% Lindo/GRASP
10x5_0	10	5	45	61	46	35.56%	24.59%
10x5_1	10	5	40	52	40	30.00%	23.08%
10x5_2	10	5	61	74	61	21.31%	17.57%
15x5_0	15	5	59	60	59	1.69%	1.67%
15x5_1	15	5	64	77	64	20.31%	16.88%
15x5_2	15	5	41	65	41	58.54%	36.92%
20x5_0	20	5	88	121	106	37.50%	12.40%
20x5_1	20	5	66	74	66	12.12%	10.81%
20x5_2	20	5	82	115	87	40.24%	24.35%
25x5_0	25	5	96	121	113	26.04%	6.61%
25x5_1	25	5	109	147	129	34.86%	12.24%
25x5_2	25	5	86	96	96	11.63%	0.00%
Efficient mean						27.48%	15.59%

4. Conclusions

Both criteria adapted to the kind of algorithm presented: the criteria were voracious in the case of the voracious algorithm (un-modifiable selection); or adaptable and random as in the case of the GRASP algorithm. In the literature there are not any GRASP algorithm that considers a double relaxation criteria at the moment of making the correspondent selections: conventional GRASP ones, were just part of a RCL list of candidates for the tasks to be executed. In 100% of the cases the result of the GRASP algorithm is better than that of voracious algorithm for high enough test instances (proportion 4 to 1, 5 to 1). In small instances it is at least equal to the voracious solution or it does not reach the same level by a very little percentage. The percentage of advantage of the GRASP algorithm confronted to the voracious algorithm is in average 5%.

GRASP algorithm get much closer to the exact solution for analyzed instances, within an average range of 5% to 9% of those solutions. In multiple cases it equals the solution, behavior that has been seen as we reduce the task-machine proportions, this is, when the number of available machine increases.

REFERENCES

- [1] G. Miller, E. Galanter. Plans and the Structure of Behavior. Holt Editorial, 1960.
- [2] M. Drozdowski. Scheduling multiprocessor tasks: An overview. *European Journal Operation Research*, 1996, pp. 215 - 230.
- [3] R. Campello, N. Maculan. Algoritmos e Heurísticas: desenvolvimento e avaliação de performance. Apolo Nacional Editores. Brasil, 1992.
- [4] M. Pinedo. Scheduling: Theory, Algorithms and Systems, Prentice Hall, 2002.
- [5] T. Feo, M. Resende, Greedy Randomized Adaptive Search Procedure. *In Journal of Global Optimization*, No. 6, 1995, pp. 109-133.
- [6] W. Rauch, Aplicaciones de la inteligencia Artificial en la actividad empresarial, la ciencia y la industria - tomo II. Editorial Díaz de Santos. España, 1989.
- [7] P. Kumara Artificial Intelligence: Manufacturing theory and practice. NorthCross - Institute of Industrial Engineers, 1988.
- [8] A. Blum, M. Furst. Fast Planning through Plan-graph Analysis. *In 14th International Joint Conference on Artificial Intelligence*. Morgan- Kaufmann Editions, 1995, pp. 1636 -1642.
- [9] R. Conway. Theory of Scheduling, Addison-Wesley Publishing Company, 1967
- [10] M. Tupia. Algoritmo voraz para resolver el problema de la programación de tareas dependientes en máquinas diferentes. *In International Conference of Industrial Logistics (7, 2005, Uruguay)* ICIL. Editors. H. Cancela, Montevideo – Uruguay, 2005, p. 345.
- [11] M. Tupia., D. Mauricio. Un algoritmo GRASP para resolver el problema de la programación de tareas dependientes en máquinas diferentes. *In CLEI (30, 2004, Perú)* Editors. M. Solar, D. Fernández-Baca, E. Cuadros-Vargas, Arequipa – Perú, 2004, pp. 129—139.

A support vector machine as an estimator of mountain papaya ripeness using resonant frequency or frequency centroid

P.B. Bro¹, C. Rosenberger², H. Laurent², C. Gaete-Eastman³, M. Fernández¹, and M.A. Moya-León³

¹ Department of Engineering Science
Universidad de Talca
Curicó, Chile
pbro@utalca.cl

² Laboratoire de Vision et Robotique - UPRES EA 2078
ENSI de Bourges - Université d'Orléans
10 Boulevard Lahitolle, 18020 Bourges cedex, France
christophe.rosenberger@ensi-bourges.fr

³ Institute of Plant Biology and Biotechnology
Universidad de Talca
Talca, Chile

Summary. Mountain papaya fruits (*Vasconcella pubescens*) were tested for firmness with a nondestructive acoustic method for 14 days after harvest. The response of each fruit was analyzed with the Fourier transform to obtain a firmness index (FI) based on the second resonant frequency and with the Short Time Fourier Transform (STFT) to obtain a spectrogram frequency centroid (FC) index. The indexes were processed with a support vector machine (SVM) learning procedure in which days since harvest was taken as the basic truth of ripeness which the measured indexes attempt to estimate. The analysis of the results demonstrate that different groupings of the days into classes to be estimated give widely varying recognition rates and that the best rates are obtained when the classes are delimited using prior knowledge.

1.1 Introduction

The objective of the research reported in the paper was to use a support vector machine to evaluate quantitatively estimations of ripeness for mountain papaya (*Vasconcella pubescens*) using different measurement techniques. The basic test involved is to estimate how many days each fruit had been stored at room temperature since harvest on the basis of each method and to compare the accuracy of each method. The fundamental goal is to predict

the future evolution of the fruit from the knowledge of an easily measured, non-destructive quality parameter.

The firmness measurement methods used in this research are based on acoustic measurements of vibrational response of the fruit to impulsive mechanical excitation. This is quite similar to the fine art of testing watermelons by thwacking them with the palm of the hand, or evaluating the quality of a used car by kicking the tires. Despite almost three decades of academic research and development, the determination of the firmness index through acoustic testing has not been widely accepted in industry. Many factors must contribute to this lag; one may be that the acoustic test is overly sensitive to variations in fruit form. One purpose of the overall research of which this report forms a part, is to investigate methods of reducing the variability of the estimation procedure using machine learning.

1.1.1 Fruit firmness

The current industry standard method for measuring the firmness of fruit is a pressure tester based on work by Magness and Taylor [12] originally developed for apples. The basic procedure is to push a cylindrical probe (typically 11mm diameter) into the pared flesh of fruit (to typical depth of 7.9mm). Tests have shown average firmness values obtained with different brands of tests that are statistically significantly different from each other, and show significant dependence of the pressure on the operator [11]. Despite its variability, the pressure test is still the technique of choice for industrial operators.

As an alternative procedure to replace the pressure test, a firmness index has been developed ([8, 1, 5]). The firmness index method is based on exciting vibration in the fruit and determining its resonant frequencies. The firmness index is defined as:

$$FI = (f_{n=2})^2 m^{(2/3)} \quad (1.1)$$

where

m – mass

$f_{n=2}$ – second resonant frequency

Cooke and Rand [6] suggest that the first resonant frequency corresponds to a spheroid mode and that the second resonant frequency corresponds to a torsional mode and developed equation (1.1) based on torsional spherical models. Terasaki *et al.* [16] observed the vibration modes of apples using speckle pattern interferometry and concluded that the second resonant frequency mode in fact of an oblate-prolate mode of spherical vibration. This result invalidates the theoretical foundation for equation (1.1) however the authors conclude that the firmness index is of practical value as stated and does not merit alteration.

The vibration induced in a fruit in response to an impulsive excitation has a character which depends highly on the time and consequently one must

assume that the standard FT is not fundamentally appropriate for analyzing the acoustic response of fruit to impulsive excitation. The short time Fourier transform (STFT) can be used [13] to determine the time varying properties of a signal. With the STFT the data is screened by a sliding window such that only a short duration of the signal is transformed before moving the window to the next portion of the signal. More formally, the FT of each portion of the signal is convolved with the FT of the window. As a result a spectrogram is obtained, a set of spectra as a function of time, also equivalent to the magnitude of the STFT.

As background to this current report, [2] analyzed mountain papaya with the acoustic method and found that the centroid of a portion of the time-frequency spectrogram gives a more robust index of fruit ripeness than does the second resonant frequency. A hypothesis of this research is that the time-frequency analysis shows the response of the fruit in a fashion which is more productive than the static resonant analysis.

1.1.2 Machine learning

The framework for machine learning in the current context is to start with a set of pairs of parameters describing fruit ripeness: $\{x_i, y_i\}_{i=1, \dots, \ell}$ where x_i describes the fruit ripeness and y_i is a quality index of the fruit. In our case y_i is the number days since harvest, the basic truth which is known, and x_i is the resonant or centroid frequency, which is measured. Our objective is to learn, on the basis of a training set $\{x_i, y_i\}$, a function f that will be able to estimate accurately the index quality on the basis of the measured parameter. The procedure is to divide a data set of parameter pairs into a learning set and a validation set. One would assume that if the learning set approaches 100% of the total data base, then the validation will also be relatively high.

We use a supervised learning framework to define the estimation functional. This is a multi-class learning problem in which the number of classes depends on the cardinality of index quality. For instance, the fruit firmness can be discretized on a period of 10 days yielding thus into a 10-class problem in a d dimension space. The multi-class problem is addressed through a polychotomy based on a *one-against-one* approach [9]. In the present case, the classes into which the fruit are to be classified are groups of days since harvest. Note that during the first few days after harvest, the change in frequencies is significant, but after about a week, the fruit is already quite mature, and the frequencies do not change thereafter. Therefore, the class grouping might not necessarily be uniformly distributed across the days.

Our machine learning algorithm for each binary problem is a 2-norm support vector machine (SVM) [7], which has already demonstrated its efficiency in other applications [10, 3, 14, 17].

In this method we look for a hyperplane in \mathcal{H} space defined as:

$$f(x) = \sum_{i=1}^{\ell} \alpha_i^* y_i K(x_i, x) + b \quad (1.2)$$

that maximizes the separation, or margin, between the hyperplane and the data points x_i projected onto \mathcal{H} . Here α_i^* are the solutions to the following optimization problem:

$$\begin{cases} \max_{\alpha_i} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (K(x_i, x_j) + \frac{1}{C} \delta_{i,j}) \\ \text{with } \sum_i \alpha_i y_i = 0 \quad 0 \leq \alpha_i \end{cases} \quad (1.3)$$

where K is the kernel associated with \mathcal{H} , $\delta_{i,j}$ is the Kronecker delta function and C is a trade-off parameter between the margin width and the number of training examples located outside the margin.

1.2 Materials and methods

1.2.1 Plant Material

Mature mountain papaya (*Vasconcellea pubescens*) fruits were collected during summer season of 2004 from commercial orchards located at Lipimavida (34°51'S; 72°08'W; 5 m ASL), on the coastal area of VII Region, Chile. Fruits that were below 5% of yellow color and free from any injuries were harvested using a random sampling method. Then fruits were randomly separated in two groups: one group was treated with 1-MCP (0.3 $\mu\text{l.l}^{-1}$) during 16 hours at 20°C in the same day of harvest (1-MCP group), while the other group was left untreated (control group). The fruit was allowed to ripen in a room at 20°C. In this paper we deal only with the control group, not the 1-MCP treated fruit.

Four fruits were randomly chosen from each group, and pressure (destructive method) and ethylene production were followed during ripening at 20°C every two days (February 2004 set). The experiment was repeated on March 2004, in which pressure was followed on each fruit in a daily base by using the non-destructive acoustic method. In order to obtain a more complete data set, the results of both trials were combined by normalizing the measurement date to be number of days since harvest.

1.2.2 Pressure and firmness measurements

Acoustic response for firmness measurement was obtained submitting each fruit to impulsive excitation, using a light hammer and piezoelectric sensor similar to the configuration of [15]. The piezoelectric film sensor (Imageco Corp., New York, USA), measuring 80mm long by 15mm wide, was glued to a 10mm thick foam pad in a plastic apple processing line cup, measuring 160mm wide by 140mm long and 40mm deep. The hammer, made of an 8mm

diameter wooden dowel, 300mm long, was given a 50g lead counter weight and balanced so as to rebound after contacting the fruit and not return to touch the fruit a second time. The experimental setup is illustrated in Figure 1.1.

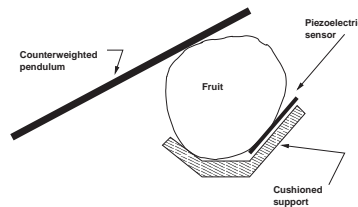


Fig. 1.1. Experimental equipment for firmness measurement.

The signal was captured to a PC running Fedora Linux, through the audio card (Creative Sound Blaster, Audigy Plus) at a 44100Hz sampling rate for 4096 samples. The signal capture software was written in C language using the Open Sound System (<http://www.opensound.com>) application program interface. The signal was captured to a file for off-line analysis, programmed using Octave (<http://www.octave.org>), an open source language and library of routines for numerical analysis and MatLab®.

First the FT was taken of each signal to determine the second resonant peak. A search procedure was programmed to pick out the frequency of the highest peak after 100Hz. This threshold was chosen so as to skip the first resonant peak which in many cases is of higher amplitude than the second resonance. The second analysis of the response signal was to perform the STFT. Since the highest frequencies were less than 2KHz, the signal was decimated by a factor of 8. A Hamming window of 71 sample width was used to obtain the spectrograms.

The SVM was applied using the libSVM library published by [4] and available on-line. The initial steps were to process the data using MatLab® to calculate the spectra, and the time frequency responses. The resonant and centroid frequencies were recorded in the format expected by the libSVM sequence of programs along with the day since harvest, on which the SVM was trained and then used for prediction. The learning set was repeatedly altered so that each part of the data was used to predict the other part of the data set. The data set was grouped into subsets, with varying sizes and numbers of members, to investigate whether size of subset affects the prediction results. Different percentages of the data were used as the learning data set to establish the improvement of recognition as a function of percentage of the data used for learning, and the grouping of days since harvest was varied.

1.3 Results

The results of the initial tests are represented by the graph in Figure 1.2. The daily spectra obtained from one fruit (Control fruit 1, February tests) are plotted with the value of the ordinate offset to distinguish between each day. The first day is at the top of the graph and later days follow sequentially below. A first resonance at about 40Hz is common for all the days, and higher resonances between 150 and 350Hz change their position and form with the day. Note that the resonance at about 150Hz can be regarded as increasing in importance relative to the resonance which begins at 300Hz and ends near 250Hz.

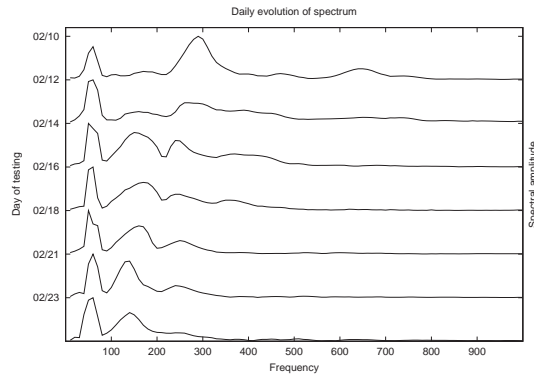


Fig. 1.2. Evolution of spectrum, fruit 1.

The FI described in the previous section is based on the second resonant frequency. In Figure 1.2 the daily evolution of this frequency is illustrated for one fruit. In order to calculate the FI, one must decide which resonant peak is to be taken as the second resonance. We have used an automated procedure which simply takes the highest peak above 100Hz as the second resonance. With this value one can visualize the daily evolution of the second resonance of the various fruit being tested.

The results of the time-frequency analysis are illustrated in Figure 1.3 where spectrograms are given for the same fruit as illustrated in Figure 1.4. The centroid used in this paper refers to the average frequency of a small portion of the spectrogram at the initial part of the signal. Figure 1.4 illustrates the evolution over time of the resonant and centroid frequencies, the two basic measurements which this paper seeks to compare.

The recognition rate for the SVM is illustrated in Figure 1.5, which includes rates for both centroid frequency and resonant frequency as the independent parameter, x_i , from which the quality index, y_i is estimated. The variation of number of sample sets and size of each set had very little effect on

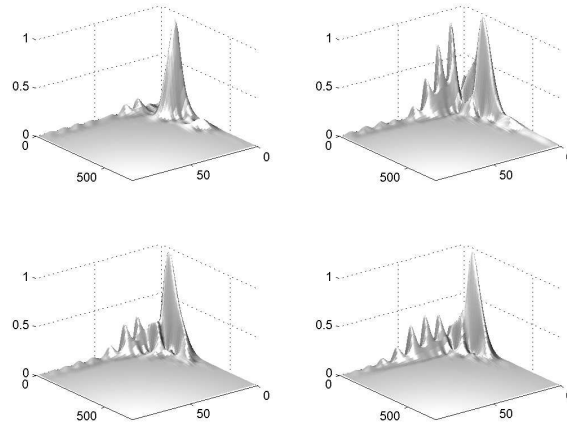


Fig. 1.3. Spectrograms control fruit 01, February data.

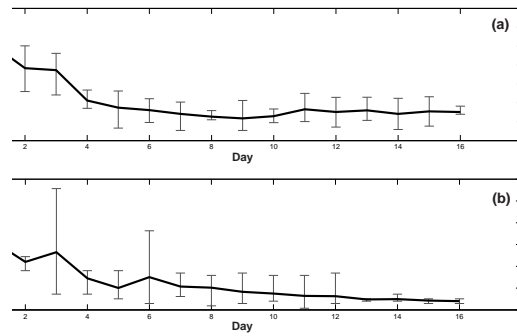


Fig. 1.4. Resonant (a) and centroid (b) frequencies, control fruit.

the recognition rate; the results presented here are for 17 sets of 5 elements each.

Several groupings for days since harvest are used:

1. Each individual day forms one class
2. Each day up to day 6 form individual classes and all days thereafter form an additional class
3. Each day up to day 6 form individual classes and all days thereafter are discarded
4. Days are grouped 1+2, 3+4, 5+6, >6
5. Days are grouped 1+2, 3+4, 5+6, all days greater than 6 are discarded

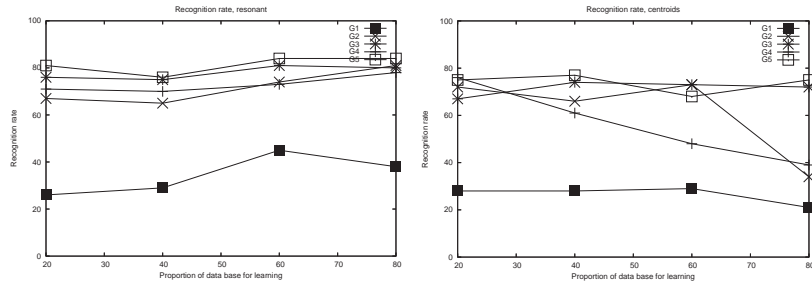


Fig. 1.5. Recognition rate for resonant (a) and centroid (b) frequencies, control fruit .

These groupings were chosen from inspection of the curves relating centroid frequency or resonant frequency to day after harvest. In both cases, after about 6 days, the frequencies no longer vary with day. This is because the fruits have reached a state of ripeness where they no longer become softer with each passing day. In fact the fruits begin to mold and the outer peel begins to form a new slightly stiffer material where the fungal growth contributes to the structural stability of the fruit.

In the concluding section of this paper we analyze the evolution of the recognition rate of the two indexes. These results are somewhat surprising and lead to careful thinking about the use of the SVM and the value of the firmness indexes.

1.4 Discussion and conclusions

The original objective of this research was to compare the FI with the FC methods of ripeness estimation, using the SVM to obtain a quantitative measure of the accurate of each measurement method. The hypothesis was that the centroid method would give a more robust and dependable prediction than the resonance method, due to the clearer functional dependance of centroid frequency on day, with respect to the dependance of resonant frequency on day, as per the conclusion reported by [2] and reflected in Figure 1.4.

This quantitative measure has been achieved in the results of this research, as the recognition rate obtained by the SVM, however it is most decidedly not the expected result, but higher for the resonant than for the centroid frequency. This was not expected since the dependance of centroid frequency on day after harvest is clearer than for the resonant frequency. However the mature fruit resonance peaks are more centralized than are the centroid frequencies of these same fruit.

An initial conclusion to be drawn from the results is that the grouping of the parameter pairs, $\{x_i, y_i\}_{i=1, \dots, \ell}$ into classes by individual days (G1) results

in a much lower recognition rate than by more general groupings. Using the knowledge that the fruit after day 6 is overripe and grouping all this fruit into one class, or by directly discarding these parameter pairs, results in significantly higher recognition rates.

A second result that was unexpected is that the recognition does not increase notably as the size of the learning data base increases. In particular for the centroid based recognition, the groupings of all the days greater than 6 into one class (G2 and G4) drop rather sadly when 80% of the data base is used for learning and only 20% is used for validation. This can be termed counterintuitive, since one would expect *a priori* that use of a higher percentage of the data base for learning would ensure a high validation rate. The resonance estimations are not significantly affected by the size of the learning data base, not improving much, but at least not getting worse. We do not have a convincing explanation for this result.

The resonant frequencies of the overripe fruit also show less variability than the centroids of the same fruit at that stage. This homogeneity may be an artifact of the technique for calculating the resonant frequency, and indeed may be the reason that the SVM predicts the day after harvest more successfully with resonant frequency than with the centroid frequency.

This research is part of an overall effort to assist the fruit processing industry produce uniform, high quality food. Future directions must include extensive, large scale testing to determine if one or another of these non-destructive indexes is most suited for classifying fruit into commercially relevant categories. One such activity will be to test a large number of fruit, as opposed to the very limited sample used in this research. As was mentioned in the materials and methods section, the initial data set in each of the trial runs consists of only four fruits, due to the equipment available for capturing ethylene gas. This represents a clear limitation of the present study; future research will concentrate on the physical, acoustic tests, and so a larger test sample may be used.

Another important goal of future activities is to associate the significance of the secondary peaks with precise physiological events in the ripening process. We have seen that these peaks rise and fall over the ripening period, confounding the second resonant technique. However the information contained in the relative importance of the peaks reflects the internal state of the fruit, which is precisely what the phytophysiologist needs to understand. In this research we have not reaped all the information made available through the time-frequency analysis, and intend to continue exploring the use the Fourier and wavelet transforms for fruit response analysis. Finally, we would like to explore other machine learning techniques which might be useful for assessing fruit characteristics with non-destructive, on-line testing.

References

1. J.A. Abbott, N.F. Childers, G.S. Bachman, J.V. Fitzgerald, and F.J. Matusk. Acoustic vibration for detecting textural quality of apples. *Am. Soc. for Hort. Sci.*, 93:725–737, 1968.
2. P.B. Bro, C. Rosenberger, H. Laurent, C. Gaete-Eastman, M.A. Moya-León, and M. Fernández. Application of a support vector machine to spectrograms for determinatino of papaya ripeness. In *Proc. Workshop on Artificial Intelligence*, Valdivia, Chile, November 2005. Sociedad Chilena de las Ciencias de la Computación.
3. Doina Caragea, Dianne Cook, and Vasant Honavar. Towards simple, easy-to-understand, yet accurate classifiers. *IEEE Conf. on Data Mining*, page 497, 2003.
4. Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
5. J.R. Cooke. An interpretation of the resonant behavior of intact fruit and vegetables. *Transactions ASAE*, 15(6):1075–1080, 1972.
6. J.R. Cooke and R.H. Rand. A mathematical study of resonance in intact fruits and vegetables using a three media elastic sphere model. *J. of Agric. Eng. Research*, 18:141–157, 1973.
7. Cristianini and J. Shawe-Taylor. *Introduction to Support Vector Machines*. Cambridge University Press, 2000.
8. E.E. Finney. Dynamic elastic properties of some fruits during growth and development. *J. of Agric. Eng. Research*, 12(4):249–256, 1967.
9. C.-W. Hsu and C.-J. Lin. A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, 13:415–425, 2002.
10. Anil K. Jain, Robert P.W. Duin, and Jianchang Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1), January 2000.
11. Eugene Kupferman and Nairanjana Dasgupta. Comparison of pome fruit firmness testing instruments. *Postharvest Information Network*, 2001.
12. J.R. Magness and G.R. Taylor. An improved type of pressure tester for the determination of fruit maturity. *USDA Circular*, 350, 1925.
13. Alan V. Oppenheim and Ronald W. Schaffer. *Discrete-time signal processing*. Prentice-Hall, 2 edition, 1998.
14. C. Rosenberger, A. Rakotomamonjy, and B. Emile. Generic target recognition. In *EUROSIP Journal on Applied Signal Processing*, pages 1613–1616, 2004.
15. I. Shmulevich, N. Galili, and D. Rosenfeld. Detection of fruit firmness by frequency analysis. *Transactions ASAE*, 39(3):1047–1055, 1996.
16. S. Terasaki, N. Sakurai, N. Wada, T. Yamamishi, R. Yamamoto, and Nevins D.J. Analysis of the vibration mode of apple tissue using electronic speckle pattern interferometry. *Transactions ASAE*, 44(6):1697–1705, 2001.
17. P. G. Wetzlr, R. Honda, B. Enke, W. J. Merline, C. R. Chapman, and M. C. Burl. Learning to detect small impact craters. *IEEE Workshop on Applications of Computer Vision*, 1(1):178–184, 2005.

FieldPlan: Tactical Field Force Planning in BT

Mathias Kern, George Anim-Ansah, Gilbert Owusu, and Chris Voudouris

Intelligent Systems Research Centre
BT Group CTO Office
Adastral Park, Martlesham Heath
Ipswich IP5 3RE, UK

{mathias.kern,george.anim-ansah,gilbert.owusu,chris.voudouris}@bt.com

Abstract. In a highly competitive market, BT¹ faces tough challenges as a service provider for telecommunication solutions. A proactive approach to the management of its resources is absolutely mandatory for its success. In this paper, an AI-based planning system for the management of parts of BT's field force is presented. FieldPlan provides resource managers with full visibility of supply and demand, offers extensive what-if analysis capabilities and thus supports an effective decision making process.

1 Introduction

BT is a leading provider of telecommunication solutions servicing customers in the UK and throughout the world. Like any other service organisation, BT is faced with the stern challenge of delivering services optimally to its customers. The effective management of resources is a fundamental and critical part of this challenge. A proactive resource management approach, i.e. an approach that provides full visibility of the service chain, that offers extensive analysis capabilities, that is automated and user-friendly, is required. [1, 2] are examples of the considerable research and development effort put into the automation of resource management.

BT's Intelligent Systems Research Center has developed a fully integrated suite of applications for the management of field force resources [3, 4]. Among other components, this suite includes:

- *FieldForecast* allows the forecasting of expected demand volumes.
- *FieldPlan* is an application for resource planning.
- *FieldSchedule* is a resource scheduling system.
- *FieldExchange* facilitates the redistribution and exchange of resources.
- *FieldPeople* is a tool for gathering and managing all people-related information such as availability and working patterns.
- *FieldReserve* is a reservation system for incoming work requests.

¹ British Telecommunications plc

The main focus of this paper is FieldPlan. FieldPlan is a planning system for the management of field force resources such as engineers and technicians. It incorporates a planning algorithm based on heuristic search that efficiently and effectively aligns the expected demand for services with the available field force supply. While doing so, resource managers can actively engage in scenario modeling and thus can analyse the problem from different perspectives in order to reach better informed decisions.

The paper is structured as follows. In section 2, the scenario for field force planning in BT is described. Section 3 is a detailed account of the FieldPlan planning approach. Initial results are presented in section 4, followed by concluding remarks in section 5.

2 The BT Field Force Planning Scenario

With regard to resource planning, one has to distinguish between three main types: scheduling, tactical planning and strategic planning. Scheduling is resource management very close to the actual time of service provision, and may look a few hours up to one or two days ahead. Tactical resource planning is short to mid-term planning as it deals with time windows of a few days up to several weeks. If planning is performed for longer time phases, i.e. up to several years, this is then viewed as a strategic planning task.

The resource planning scenario to address here is a tactical one. A workforce of field engineers, the resources, have to be optimally deployed to serve expected demand in form of jobs. This means that plans have to be generated for the field force, on a per day or per week basis, for up to 12 weeks. FieldPlan has to deploy up to 300 engineers to fulfil up to 20,000 job requests in a single planning run. The main planning objective is to optimally utilise the field engineers to complete jobs while reducing operational costs. Jobs and engineers are characterised by the attributes listed below:

Job	Engineer
geographical location	list of geographical locations, with preferences
skill	list of skills, with preferences
product	list of products, with preferences
time window	availability over time
type (actual or forecasted)	type (standard, loan-in, contractor, etc.)

Each job is either an actual job, i.e. a real customer demand, or is forecasted by the FieldForecast system. A job involves the provision, repair or cease of a product at a particular location and requires a specific skill. This should be accomplished within a given time frame. The importance of a job indicates the value of the respective work to the business.

Engineers, on the other hand, possess sets of skills, can provide various products and are flexible in their choice of working area. Preferences for particular areas, skills and products² are possible. Different types of engineers can be deployed, e.g. standard BT engineers or contractors. The engineers show varying daily availabilities, in terms of working hours, during the planning time window.

General importance and productivity rates of skills and products are also known.

In contrast to scheduling systems [5] which assign engineers to particular jobs, the goal here is to determine the best arrangement of engineers in terms of areas, skills and products. For every time period within the planning horizon, each engineer has to be assigned to an area or a set of areas, a skill or a set of skills, and a product or a set of products. Furthermore, the respective numbers of engineer working hours have to be established. The decision about what jobs engineers will actually work on is left to a scheduler at a later point.

Results of the planning process are presented in two forms: capacity and deployment plans. A capacity plan is a coarse-grained summary of the service situation purely in volumes: how many jobs are expected to be cleared/uncleared per area, skill and product; how many engineers are available and utilised per area, skill and product; in short, can the demand be met with the available supply. The deployment plan refines this information by giving explicit area, skill and product deployment recommendations together with expected utilisation percentages for each engineer.

3 The FieldPlan System

In this section, the planning algorithm and its main components are described and important issues are discussed. A schematic outline of the planning approach is given in figure 1.

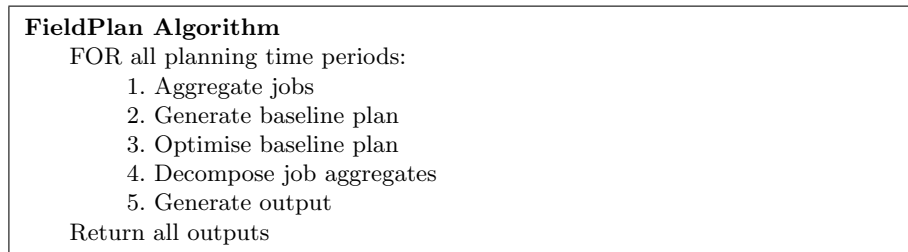


Fig. 1. The FieldPlan planning algorithm

² In the current version of FieldPlan in BT, engineers are deployed by default to all products.

Planning Time Window and Periods The process of resource planning is performed for a given time frame. This time window is usually further divided into shorter intervals, and plans have to be generated for all individual stages. A typical tactical example is the planning for two weeks on a per day basis.

The FieldPlan system captures this notion by introducing planning time periods. A planning time period is defined by a start and end time and thus represents a time interval. The overall planning time window is, consequently, a sequence of non-overlapping time periods. By basing the core planning mechanism in FieldPlan purely on the generic concept of time periods, we are able to express the algorithm without any reference to actual time intervals like days or weeks.

A central characteristic of our planning algorithm is the iterative construction of the overall plan out of partial plans. For each individual time period, a partial solution is constructed. Rather than constructing a single plan for the whole time window in one step, a number of steps yields a number of partial solutions which are finally combined to form the complete plan.

Initial Planning Time Period The first planning time period represents the current time period, i.e. it contains the current time at the point of planning. It differs therefore from all subsequent time intervals in two aspects:

- At the point of planning, engineers might already be working. Consequently, they are already assigned to specific areas, skills and products for the first time period. These restricted choices have to be considered when evaluating the future course of the first period.
- Furthermore, parts of the initial time period might have already past at the point of planning. This must be reflected by reduced working hours of the engineers. If, for instance, 50% of time period 1 have already past then the available engineer capacities must be reduced by half.

Job Aggregation and Decomposition In contrast to scheduling systems, FieldPlan does not assign engineers to work on particular jobs. The aim of tactical planning is of a more general nature, namely to decide what kind of engineers have to be deployed to which location to fulfil what kind of jobs. This generalisation allows the planning process to work with job categories rather than single jobs. Hence our planning system merges single jobs into more general objects referred to as job aggregates and bases all planning decisions purely on these aggregations. The advantage of this approach is clear: reduced complexity while maintaining a sufficient level of granularity. Planning becomes easier, faster and more scalable.

In our scenario, a job aggregate is described by the following information:

Job Aggregate
geographical location
skill
product
state
volume

This means that jobs that match in area, skill, product and state are aggregated. Instead of a time window, job aggregates possess a state: backlog, current or future. Backlog job aggregates should have been finished before the current planning time period, current aggregates should be completed within the current time period, and future job aggregates have a later target completion. The volume indicates the number of aggregated jobs.

In the main planning algorithm, all jobs are initially aggregated during each time period. After generating the baseline plan and optimising it, the cleared and uncleared job aggregates are decomposed back into cleared and uncleared jobs.

Objective Function The objective function is the central instrument of evaluating the quality of a particular resource deployment. By assigning a numeric value to each such deployment, comparing candidate solutions and thus selecting the better one is made possible. The compositions of the FieldPlan objective is illustrated below.

For a given job aggregate $\mathcal{A} = (area, skill, product, state, volume)$, the *job aggregate clearance score* $jacs(\mathcal{A})$ is defined as

$$jacs(\mathcal{A}) = importance(\mathcal{A}) \cdot \frac{volume}{productivity(skill, product)}. \quad (1)$$

It is a measure for the value of clearing the job aggregate. The formula normalises the volume with regard to the productivity: the longer work takes, the higher is the score. The score is also better if the job aggregate is more important. In our current implementation, $importance(\mathcal{A})$ is calculated as $importance(skill) + importance(state)$, but more complex measures that consider area and skill as well are possible.

For an engineer \mathcal{E} with assigned job aggregates $\mathcal{A}_1^{\mathcal{E}}, \dots, \mathcal{A}_n^{\mathcal{E}}$, the *engineer clearance score* $ecs(\mathcal{E})$ is given as

$$ecs(\mathcal{E}) = \sum_{i=1}^n jacs(\mathcal{A}_i^{\mathcal{E}}). \quad (2)$$

While the engineer clearance score only evaluates the (quality of) service delivered by an engineer, the *engineer score* $es(\mathcal{E})$ takes also service costs into account as it considers the actual deployment of the engineer:

$$\begin{aligned}
es(\mathcal{E}) &= ecs(\mathcal{E}) \cdot \\
&\quad area_penalty(\mathcal{E}) \cdot skill_penalty(\mathcal{E}) \cdot product_penalty(\mathcal{E}) \cdot \\
&\quad utilisation_penalty(\mathcal{E}) \cdot technician_type_penalty(\mathcal{E}). \tag{3}
\end{aligned}$$

Above penalty functions return values between 0 and 1. The original engineer clearance score is reduced by this formula if the engineer is not assigned to the preferred area, skill or product. Further penalties might be applied in case of poor utilisation or if the engineer has to be brought in as additional workforce like contractors or on overtime.

The overall *objective score* obj of a deployment of the engineers $\mathcal{E}_1, \dots, \mathcal{E}_n$ can be calculated as

$$obj(\mathcal{E}_1, \dots, \mathcal{E}_n) = \sum_{i=1}^n es(\mathcal{E}_i). \tag{4}$$

This objective is a balanced measure of the quality of service delivered by all engineers and the incurred service costs.

Baseline Plan Generation The initial baseline plan is generated by assigning all engineers to their default choices, i.e. to their preferred areas, skills and products. The least flexible engineers in terms of available areas, skills and products are deployed first to ensure that even highly constrained engineers are able to pick up work. Assigning engineers to such choices involves the assignment of matching uncleared job aggregates according to their capacities.

Plan Optimisation The strategy employed in FieldPlan to optimise an initial baseline plan is a hill-climbing algorithm. The current plan is iteratively modified in small steps which are referred to as moves. Each new modification of the solution is characterised by an improved, i.e. higher, objective function evaluation. Three types of moves are considered by FieldPlan:

- *Move-to moves*: Deployments of single engineers are altered by assigning them to new areas, skills and/or products.
- *Swap moves*: The deployments of two engineers are exchanged.
- *Replacement moves*: This move type involves a chain of two or more engineers. The first engineer replaces the second engineer, the second engineer replaces the third, and so on. The final engineer is thus freed and can be moved elsewhere. Currently, only chains of length two and three are considered.

The optimisation routine repeatedly applies the three types of moves to the engineers. As long as improvements are found, the process continues. Only when none of the considered moves offer any improvements anymore, the optimisation is halted as a local maximum is reached.

Heuristic search algorithms such as Simulated Annealing [6], Tabu Search [7], Genetic Algorithms [8] or Guided Local Search[9] employ strategies to escape such local extrema and to improve the generated solution even further. Although these advanced search methods can be easily incorporated in FieldPlan using the defined objective function and move operators, we use the more basic

hill-climber due to computation time restrictions. As interactivity is application critical, fast computation and response times are of uttermost importance. The less time intensive hill-climbing approach can meet these requirements and is thus an appropriate choice.

Output At the end of each optimisation cycle, the compiled planning solution is analysed by FieldPlan to provide resource managers with vital information on the state of the supply chain. In addition, FieldPlan computes data that can be utilised as inputs by further systems. The following list gives an overview of the different categories of analysis:

- *Capacity plan:* On the one hand, this plan analysis provides data about the available capacity in the form of engineers. On the other hand, it lists required capacity in form of jobs. Resource managers are able to see how well both match, how many cleared and uncleared jobs can be expected and how well the engineers will be utilised.
- *Deployment plan:* This output makes explicit area, skill and product recommendations for each engineer for all time periods. This deployment information can consequently be used as direct input to a scheduling system.
- *Under-utilisation:* FieldPlan identifies engineers who are not fully utilised. If the utilisation level fall below a certain threshold, reasons for the poor efficiency are compiled and recommendations for improving the utilisation are given.
- *Training:* By analysing the demand for each skill, bottlenecks can be identified and according training recommendations can be given.
- *Resource redistribution:* A resource redistribution system like FieldExchange aims to balance resources across a wider geographical area. Resource shortages and surpluses indicated by FieldPlan can be used as a data feed.
- *Appointment booking:* The analysis of available engineer capacity can be used by appointment booking systems.

Plan Types and Planning Options FieldPlan distinguishes three levels of plan optimisation: baseline plans, optimised plans and user-specified plans. The main differentiator is the level of flexibility the planning mechanism is given in deploying the resources. On one side of the scale, baseline plans do not offer any flexibility at all. All technicians are restricted to their default areas, skills and products, and no overtime is available. On the other side of the scale, optimised plans do not restrict technician deployments in any way, e.g. technicians can be moved to any of their areas, skills and products, and allow the full utilisation of all available overtime. User-specified plans enable resource managers to vary the level of flexibility between the two extrema via a set of planning controls. The most important parameters for such what-if analysis in FieldPlan are:

- The amount to penalise engineers, in terms of the engineer score, for not being deployed to their default areas, skills and products can be varied.
- Productivity rates can be set by the user.
- The importance of skills can be varied.

- The importance of job aggregate states can be varied.
- The usage of overtime can be switched on or off.
- A minimum target utilisation can be specified.
- Jobs can be excluded from the planning by area, skill, product and type.
- Resource managers can choose to not deploy engineers to backlog, current or future work.

All these options are calendarisable, i.e. can be set independently for each planning time period. They provide resource managers with rich scenario modeling capabilities and thus support the decision making process.

4 Results

We present planning results for four real-life scenarios for a single planning time period and compare the performance of FieldPlan’s baseline plans (BL), plans produced by FieldPlan’s hill-climbing method (HC) and plans generated by standard Tabu Search³ (TS) and Simulated Annealing⁴ (SA) implementations which are based on the same neighbourhood structure and objective function as the hill-climber.

For each of the test cases, the following table shows the objective score for each plan type, together with the number of area/skill moves⁵, total number of cleared jobs and number of cleared backlog/current/future jobs. Backlog jobs had the highest importance in all four scenarios, followed by current jobs and future jobs. The number of deployed engineers ranged from 35 to 150.

We can observe that the baseline plans have the lowest objectives in each of the scenarios. The hill-climber produces improved objective scores, but the best results are always achieved by Tabu Search and by Simulated Annealing. The improvements from baseline plans to hill-climber plans are much more dramatic than the improvements from hill-climber plans to Tabu Search/Simulated Annealing plans.

While the baseline plans disallow any kind of area/skill moves, the other algorithms redeploy engineers in terms of area or skill. This increased flexibility results in better plans. The hill-climber, Tabu Search and Simulated Annealing produce plans which show higher total job clearance rates in all scenarios except case two. However, notably more high importance jobs (backlog) are cleared by those algorithms in that particular scenario.

Overall we can state that plan optimisation through hill-climbing, Tabu Search and Simulated Annealing leads to the clearance of more and more important jobs in our experimental study with the latter two approaches consistently

³ The Tabu Search algorithm employs aspiration, marks technicians who are involved in a move as taboo for $\frac{\text{number of engineers}}{5} \pm 2$ cycles and runs for 500 iterations.

⁴ The Simulated Annealing method uses an exponential cooling scheme and runs for 40,000 moves (accepted and refused).

⁵ Engineer deployments away from default choices.

Plan Type	Objective	# Moves	#Cleared Jobs	#Backlog	#Current	#Future
BL	27.91	0	227	88	139	0
HC	30.44	3	236	100	136	0
TS	31.07	4	245	101	144	0
SA	31.09	3	246	100	146	0
BL	112.40	0	776	405	222	149
HC	124.41	17	746	489	233	24
TS	125.65	23	746	525	204	17
SA	126.11	24	751	517	227	7
BL	98.85	0	781	351	244	186
HC	116.44	22	822	442	271	109
TS	117.57	28	815	464	266	85
SA	117.56	27	803	459	268	76
BL	107.55	0	793	354	323	116
HC	119.91	24	822	516	235	71
TS	122.02	33	841	568	194	79
SA	122.85	36	834	571	189	74

performing best. But these two algorithms come with higher computational costs. Both Tabu Search and Simulated Annealing take about 50 times longer than the hill-climbing method. In real-life scenarios, the improvements achieved by Tabu Search and Simulated Annealing are not significant enough to justify their much slower response times. Computation times of more than 5 minutes on typical server hardware for seven day plans are not acceptable for an interactive planning system. FieldPlan therefore employs the hill-climbing algorithm for plan optimisation to balance between performance and computation time.

After a successful trial period, the FieldPlan system is currently used nationwide (UK) for the deployment of 4,000 engineers of a BT sub-division. Reports indicate improvements in all their key performance measures, examples of which include:

- 8%-14% more engineers are assigned a job first thing in the morning, i.e. do not have to wait. The respective travel time has been reduced from 95min to 85min due to the better initial placement of engineers.
- Because of automation gains with FieldPlan, manual intervention by controllers has been limited to 18% (down from 31%).
- Quality of service has been improved, e.g. 1.1% more business provision jobs are completed on time.

The application of FieldPlan to a wider field force of 25,000 BT engineers is currently in preparation. Deployments are expected to start in February/March 2006.

5 Conclusions

The Aberdeen Group [10, 11] argues that service organisations have to optimise and automate the operations of their field services in order to improve efficiency, profitability and customer satisfaction. Without the constant drive for such improvements, survival in the highly competitive market places of the 21st century seems impossible. BT, as a provider of telecommunication services, faces these challenges every day.

In this paper, we have introduced FieldPlan, a tactical resource planning solution for the optimisation of parts of BT's field force. We have outlined the resource management scenario, have described the core planning mechanism and have discussed the extensive scenario modelling options. Initial results of the application of FieldPlan to the management of 4,000 field engineers are promising.

The current development work focuses on the large-scale application of FieldPlan to 25,000 BT field engineers. Future research will include the development of a generic planning framework and its extension to strategic planning scenarios.

References

1. Azarmi N. and Smith R. (1995) Intelligent Scheduling and Planning Systems for Telecommunications Resource Management. In *BT Technology Journal*, 13(1):7–15.
2. Laithwaite R. (1995) Work Allocation Challenges and Solutions in a Large-Scale Work Management Environment. In *BT Technology Journal*, 13(1): 46–54.
3. Voudouris C. et.al. (2005) ARMS: An Automated Resource Management System for British Telecommunications plc., In *European Journal of Operational Research*.
4. Owusu G. et. al. (2003) ARMS: Application of AI and OR Methods to Resource Management, In *BT Technology Journal*, 21(4):27–32.
5. Lesaint D., Voudouris C. and Azarmi, N. (2000) Dynamic Workforce Scheduling for British Telecommunications plc., In *Interfaces*, 30(1):55–66.
6. Kirkpatrick S., Gelatt C. D. and Vecchi M. P. (1983) Optimization by Simulated Annealing, In *Science*, 220(4598):671–680.
7. Glover F. (1989) Tabu search: Part I. In *OSRA Journal on Computing*, 1(3):190–206.
8. Holland J. H. (1975) *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Michigan.
9. Voudouris C. and Tsang E. (1999) Guided Local Search and its application to the Travelling Salesman Problem. In *European Journal of Operational Research*, 113(2):469–499.
10. Aberdeen Group Report (2004) *The Field Service Optimization Benchmark Report - Tapping the Service Supply Chain for Profit and Competitive Advantage*.
11. Aberdeen Group Report (2005) *Optimising Field Service to Achieve Profitability Goals*.

An Agent Solution to Flexible Planning and Scheduling of Passenger Trips

Claudio Cubillos¹ and Franco Guidi-Polanco²

1 Escuela de Ingeniería Informática, Pontificia Universidad Católica de Valparaíso, Av. Brasil 2241, Valparaíso, Chile
claudio.cubillos@ucv.cl

2 Escuela de Ingeniería Industrial, Pontificia Universidad Católica de Valparaíso, Av. Brasil 2241, Valparaíso, Chile
fguidi@ucv.cl

Abstract. This work presents the MADARP agent architecture, devoted to the planning and scheduling of trip requests under a dynamic scenario within the context of passenger transportation systems. The architecture provides a set of base agents that perform the basic interface, planning and support services for managing different types of transportation requests by using a heterogeneous fleet of transport vehicles. The architecture was used to implement three planning models by extending base agents' behaviours. The results obtained for a set of 20 scenarios is analyzed.

1 Introduction

The field of passenger transport systems has received an increasing attention as citizens require more flexible transportation alternatives in their cities. As response, new alternatives to satisfy the transport demands of citizens are being conceived [1].

Changes in transport requirements in European citizens have brought the opportunity to create new services aimed to fulfill special transportation demand in addition to regular population mobility services. Their objective is to satisfy personal transportation requests at relatively low costs, thanks to an integrated planning with the use of the different available resources on transport networks.

From a technological perspective, the recent advances in network systems together with the low cost of the processing power have move us to the era of distributed systems and ubiquity. In this trend, the integration, transparency and interoperation among heterogeneous systems are a must. Hence, the multiagent paradigm [16] appears as a promising technology, capable of tackling these newer requirements in an efficient and sustainable way.

This work presents the MADARP architecture, devoted to the implementation of flexible passenger transportation systems. It provides agents which implement the basic planning and scheduling functionality for processing transport requests coming from different kinds of users and by considering a heterogeneous fleet of transport vehicles.

2 Transportation Requirements

From a mathematical point of view the transportation problem involved corresponds to the dynamic version of the Dial-a-ride Problem (D-DARP) known to be NP-hard. For this reason all of the commercial solutions and most of the research are focused in heuristic solutions.

Clients commonly specify transport requests with a pick-up and delivery place. They also indicate time windows, that is, time intervals within which the client has to be picked-up at the origin node and delivered at the destination node. Moreover, the requests can include further descriptions of the desired service like type and number of places, shared or exclusive use of the vehicle, wheelchair place use and any other complementary services.

Besides, the passenger transportation system we are tackling considers heterogeneous fleets of vehicles, composed by busses, minivans, vehicles for disabled people, taxis, among others. These vehicles may have diverse characteristics such as: limited passenger's capacity and availability time-periods along the day, an specific area to cover, types of seats, low floor, wide access, no stairs or complementary services like Bar, WC, air conditioning and bicycle transport among others. These properties usually affect the client's comfort and consequently their perception of the received transport service.

In addition, a dynamic scenario is considered, in which the vehicle progress is monitored; clients can modify or cancel their trip requests, vehicle delays can occur, clients may not show up at the pickup place and vehicles can breakdown, all of them involving the re-scheduling of the trips and their management.

2.1 Related work

A software system for D-DARP was proposed by Horn [7]. The optimization capabilities of the system are based on least-cost insertions of new requests and periodic re-optimization of the planned routes. Finally, Coslovich et al. [4] have addressed a dynamic dial-a-ride where people might unexpectedly ask a driver for a trip at a given stop by using a two-phase method and a neighborhood of solutions generated off-line.

Newer research tackling the dynamic problem tends to use a distributed market-based philosophy based in the Contract-Net Protocol (CNP) (see [3], [5], [6] and [11]). The MARS System [6] and the TeleTruck [3] approach use the Extended Contract-Net Protocol (ECNP) with Simulated Trading improvement for dealing with dynamics and uncertainty in a transportation scheduling problem.

Soft computing has also been applied to the transport domain with the use of genetic algorithms (GA) for the optimization of the assignment (see [8] and [15]) and systems based on an ant-colony as reported in [12]. Teodorovic and Radivojevic [14] have later studied a generic version of the dynamic DARP using fuzzy logic for the travel times, as well as Kikuchi and Donnelly [9].

Finally, agent-based systems are presented in [5], [10] and [13]. All of them make use of the CNP for the assignment of client’s rides. In addition, [10] uses a stochastic post-optimization phase to improve the result initially obtained. It works in a similar way to the simulated trading. In [13] is presented the Provisional Agreement Protocol (PAP), based on the ECNP and de-commitment. Its improvement is to allow biddings for partial routes and overcomes the Eager Bidder Problem of the CNP, that is, the contractor commits to the bid even though the bid has not been granted and hence cannot make the same bid to another.

3 The Agent Architecture

The agent architecture is built-up over the Jade agent platform [2], which provides a distributed environment organized in containers where agents can work, communicate and migrate within them. Figure 1 shows the MADARP agent architecture [5] which shows four layers that group the agents and structures according to the functionality provided. The Interface layer connects the system with the real world; the Planning layer performs the trips processing; and the Service layer provides different complementary functionalities. At the bottom the Service Ontology provides a means to integrate and make interacting the different agents and actors from the upper layers in a transparent and coherent way.

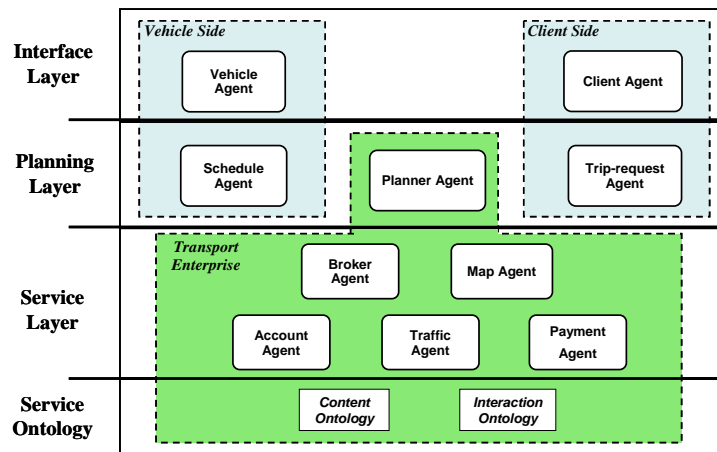


Fig. 1. The multiagent transportation architecture.

Figure 1 shows also the three main actors involved in the transportation chain: vehicles, clients and the transportation enterprise; each of them modeled in terms of

agents. Consequently, each vehicle actor is represented by a Vehicle agent and a Schedule agent. In a similar way, each client is characterized by a Client agent and a Trip-request agent. In both cases, the pair of agents is tightly coupled as they are modeling different aspects of the same real entity. The third actor is the transport enterprise, which is built up by a series of agents and structures that provide support to diverse services related with the planning and control of the passenger transportation service provided.

The routing and scheduling functionality provided by the architecture is based on the contract-net protocol (CNP) plus a possible negotiation phase. The interaction among the planning agents is as follows (see Figure 1): First, each transportation request coming from a Client is received by the corresponding Trip-request agent of the couple, which asks the Planner to process it. Next, the Planner processes the request first by obtaining from the Broker agent the vehicles that match the required profile, and then by making a call for trip-proposals to all the corresponding Schedule agents (call for bids in contract-net) that represent the different vehicles of the considered fleet. They send back their proposals and the Planner selects the most suitable alternatives among the received trip proposals by applying filters and starts a negotiation process with the client (through its Trip-request agent). After arriving to agreement the Planner tells the Schedule agent that won the proposal to add the trip to its actual schedule and tells the others their proposal rejection.

Upon differences in the planning (due to breakdowns, traffic jam, etc) the Schedule agent re-plans. In the case of having an infeasible trip request (mainly due to the time-window restrictions), it informs the Planner agent about the situation. The Planner makes a call for trip-proposals to try reallocating the request in other available vehicle. In any case, the result is informed to the corresponding Trip-request agent, which depending on its degree of autonomy will process the alternatives and take a decision or will inform the client about the change. This change may imply a different vehicle processing the trip only or also a delay or an anticipation of the pickup and delivery times defined previously. This default planning implementation can be modified or extended by overwriting the set of behaviours of the different base agents, which are detailed in the following.

3.1 Client Side

Individual clients and their requirements are captured by Client and Trip-request agents. Together they provide full communication and interoperability of the real end user with the transportation system. The Client agent is in charge of providing a personalized user interface while the Trip-request manages the process of requesting the service, its characteristics and the decision making required.

The interface provided by Client agents should be adaptable to the different devices (cell phones, PDAs or PCs). In addition, the Client agent is responsible for capturing all the client's requirements not only concerning the desired type of transport service but also his preferences upon contingency situations (e.g. delays, traffic jams, deviations, etc). The Trip-request takes these requirements and preferences to act on behalf of the real client during the whole process. Depending

on the degree of autonomy provided by the client, the Trip-request can act as a personal trip assistant or simply as a mere proxy of the client decisions.

This agent contains three base behaviours. The `Schedule_me_Behaviour` is a one-shot behaviour which in its default implementation is in charge of receiving the desired transport service from the real client through the client agent that acts as interface. The message contains the request profile that is then forwarded to the planner inside a trip request message.

The `Negotiate_Behaviour` processes the subsequent messages coming from the planner in order to arrive to an agreement. This depends on the underlying negotiation protocol implemented by both, the Trip-request and the planner agents. The default implementation for this behaviour receives a list with filtered proposals which are evaluated by using the client's utility function.

The `Send_status_Behaviour` is a cyclic behaviour performing a utility service. It turns back to the sender the status of the request being treated.

3.2 Vehicle Side

Each real vehicle is represented by an agent couple, the Vehicle and the Schedule agents. They provide interoperability between the vehicle they represent and the transportation system to which they belong to. The Vehicle agent plays an interface role, providing the vehicle and its driver with a communication channel with the rest of the transportation system. Through it, the driver is able to communicate along the journey about any contingency that could arise. The Schedule agent is in charge of managing the vehicle's route and processing any new request for client transportation.

The Vehicle agent contains two behaviours. The `Register_Behaviour` performs the registration of the vehicle with the broker agent. Therefore, it sends a subscribe message to the broker containing a Service Profile.

The `Inform_event_Behaviour` is the core behavior as it enables the agent to inform about the status of the vehicle and its route advancement. It sends an inform message to its corresponding Schedule agent containing an event description, such as the vehicle arrival to a pickup/delivery place, the presentation or no presentation of the client at the predefined stop, a vehicle malfunction, an emergency, the vehicle deviation due to an accident, traffic jam, etc.

Base Schedule agents implement two cyclic behaviours. The `Evaluate_trip_Behaviour` evaluates the insertion of a trip (client) in its current schedule. The default implementation listens to the calls-for-proposals coming from the Planner. In case of not being committed by that time with other call (not already finished) it will prepare a proposal, otherwise it will answer back with a refusal message. When preparing the proposal (profile), the behaviour decodes the client's Request Profile description attached in the call's content and evaluates the trip inclusion in the vehicle route.

The `Wait_proposal_answer_Behaviour` is much coupled to the previous one, as it process the planner's answer with respect to the formulated proposal. For this, the default behaviour checks the planner's message to see if the proposal has been accepted or rejected. In case of an accepted proposal, the behaviour uses the generic interface to insert the trip in the vehicle's route.

3.3 Transportation Enterprise Side

The transportation service role is mainly carried out by the Planner agent acting as a front face to Trip-requests and Schedule agents. The Planner has seven behaviours.

As its name says, the `Process_request_Behaviour` processes the incoming Schedule-me messages of Trip-request agents. The agent creates a registry of the new request to add to its list. It also decodes the Request Profile contained in the incoming message, and sends a query message to the broker asking for the vehicles that match the requirements contained in the Request Profile.

The `CallForProposals_Behaviour` receives the broker's answer containing the list of agents that match the desired service described in the Request Profile. The behaviour decodes the list and sends a call-for-proposal message to all the corresponding ScheduleAgents contained in the list. In case of a failure message from the broker, the behaviour will forward it to the corresponding Trip-request agent.

The `Process_proposal_Behaviour` receives the answer messages from the ScheduleAgents. These messages carry in their content the trip proposals (Proposal Profiles). The behaviour checks if all ScheduleAgents have answered back to the call. If that is the case, then a `StartNegotiation_Behaviour` is instantiated and activated in the agent.

The `StartNegotiation_Behaviour` is a one shot behaviour that can start in two ways; by a `Process_proposal_Behaviour` that received the last pending answer to the call or by a `Request_timeout_Behaviour` triggered by the call deadline. The behaviour gets all the proposals received for a given trip request and applies them the filters (if any) contained in its policies list. This will result in discarding some of the proposals. After the filter process, the behaviour starts the negotiation procedure with the Trip-request agent by sending a message to it.

The `Process_client_choose_Behaviour` also depends on the implemented negotiation protocol. The behaviour receives the Trip-request answers and sends counter proposals until a deal is obtained or the protocol finishes. In any case, the result is forwarded to the involved ScheduleAgents by sending an accept proposal message to the winner and a reject proposal message to the rest.

Finally, the `Process_schedule_confirm_Behaviour` ensures that the winning ScheduleAgent has committed to the transport request. The default behaviour receives the winner's answer and forwards it to the corresponding Trip-request agent.

Besides the Planner Agent, there is a whole set of service agents collaborating to give support to the different required functions, such as the matching of request to vehicles, the geographical data access, the accountability of the transactions and the service payment among others. From them, the most critical ones from the planning and control point of view are the Broker and the Map agents.

The Broker is the one in charge of carrying out the matching of transport requests with available vehicles. For that is able to manage service descriptions coming from both sides, understand their semantics and perform the search.

The Map agent represents the geographic area being considered where it can be a zone, a city or a part of it. The Map provides the enterprise with a series of

information regarding the actual zone being covered such as localization of addresses and stops, street names and distances between localizations, among others.

4 Concrete Planning Systems

Three models of transport planning system were implemented in order to test the agent-based architecture. These were: a centralized, a market-based (decentralized) and a mediated one. For each model, the architecture's base agents were extended and modified. These are explained in the following.

The *Centralized Model (C)* considers the optimization of the global utility for the system. It pursues the minimization of a disutility function of the fleet operator (number of vehicles required, fixed and variable travelling costs) and the served users (effective waiting time, effective ride time). For this to happen, the behaviour of Schedule agents is overwritten in order to consider the vehicle utility function plus the client's one.

The implemented *Decentralized Model (D)* is an approach based in the contract-net (CNP) under self-interested agents. Therefore, Vehicle agents pursued the optimization of the travelling costs (utility function with total slack time and total travel time) and Client agents were oriented towards the maximization of the perceived service quality (utility function with excess travel time and waiting time).

The *Mediated Model (M)* takes advantage of the mediation role of the Planner by filtering the received proposals. The mediated model involves a two-phase planning: First, a Call-For-Proposals (CFP) started by the Planner and answered by the Schedule agents and then a negotiation process that pursues an agreement between the Planner and the client. This two-phased model offers a major difference; the Planner with its filtering and negotiation policies performs a mediation role. It implements the partial centralization of this mixed approach, getting solutions closer to the global optimum when compared with complete decentralized models.

In practical terms the steps are implemented as in the previous approach (the decentralized one). It only changes the Planner role. In this implementation the Planner does apply a filtering policy to the list of received proposals.

4.1 Results

Here are presented some of the results obtained with the 3 concrete models. All the tests considered the same geographical net and 20 demand scenarios with 50 trip requests each, distributed uniformly in a two-hour horizon. For each demand scenario 25 runs were done.

The tests on the three models (see Figure 2) have shown that on average the Mediated model is able to provide results in the gap between the Centralized and Decentralized models.

In general terms, the decentralized model will provide better results for the clients, in terms of both, Waiting Time and Excess Ride Time as these are variables that measure the solution quality from the clients' perspective. In fact, when looking at Figure 2 the Decentralized values are the lowest almost for all the scenarios, while

the Centralized values correspond to the highest ones. This is explained because this model provides solutions that are better for vehicle operators and worse for clients.

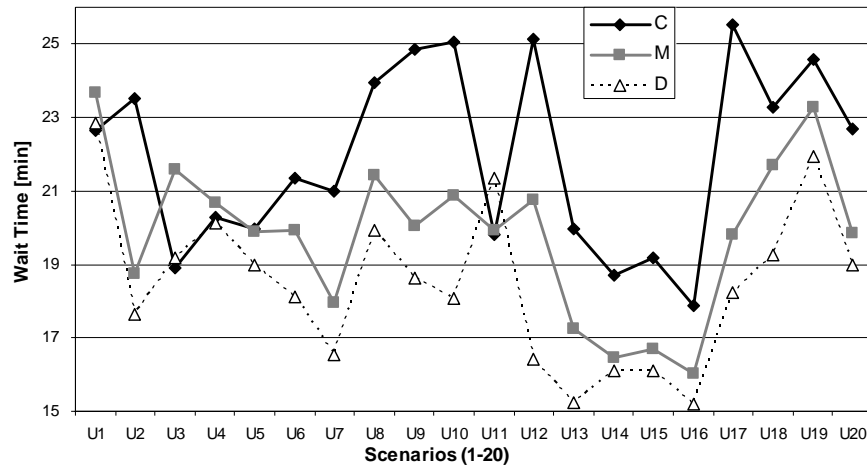


Fig. 2. Graph showing the wait time obtained for the 20 scenarios under the three planning models.

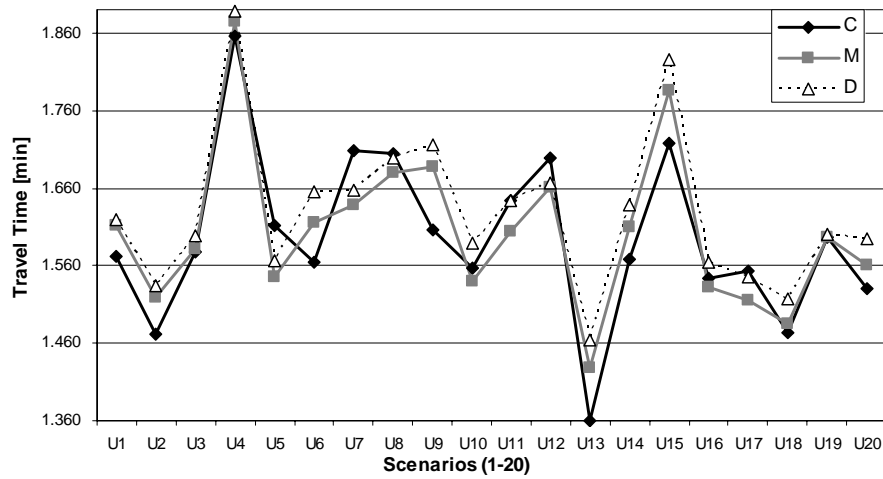


Fig. 3. Graph showing the travel time obtained for the 20 scenarios under the three planning models.

In the same Figure 2 we can appreciate that the Mediated model is just in between, providing almost in all scenarios a middle value for the wait time measure. A similar thing happens with the other client measure considered in the tests, the excess ride time.

As already mentioned, the Centralized approach tends to provide results that in terms of preference, tend to benefit more the vehicle operators rather than clients, which is just the opposite way for the decentralized model. Therefore, when

comparing the results of values regarding performance measures of the vehicles' operators, the situation is inverted. For example, in the case of the travel time, the values corresponding to the Centralized model are the lowest for almost all cases as Figure 3 shows. In fact, within the 20 considered scenarios in only five occasions the centralized was the highest among the three alternatives.

In a similar way, the Decentralized approach behaves providing the highest values for the travel time. Again in most scenarios (16 out of 20) the travel time from the decentralized model was the highest among the three alternatives. The interesting thing is that the Mediated model provides results just in between or even below the others in all 20 scenarios. Analogous results are obtained for the other operators' performance measure considered in the test, the slack time.

Finally by looking at Figure 4 that shows the average total cost of the three models on the 20 scenarios, it is possible to see that the Mediated model provides in-between or lower costs in all the 20 scenarios.

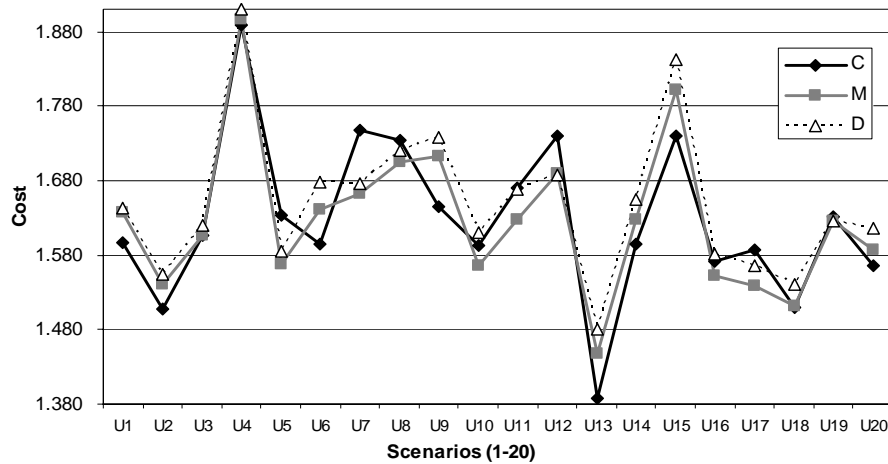


Fig. 4. Graph showing the cost obtained for the 20 scenarios under the three planning models.

To sum up, the analysis shows that the Mediated model gives better results for the clients, in terms of both, Waiting Time and Excess Ride Time when compared with the Centralized model. On the other hand, when compared with the Decentralized model, it gives better results from the vehicle's (or operator's) viewpoint, this in terms of Travel Time and Slack Time.

5 Conclusions

The MADARP architecture for implementing agent-based passenger transportation systems was described. The layered model enforces the system reutilization and maintainability, as more low level and general services (provided by JADE) are decoupled from the domain specific ones (base agents plus ontology). By extending the base agents it is possible to obtain an ad-hoc concrete planning system. MADARP

provides a transparent and flexible way to integrate users, vehicles, and support-service providers into a single architecture. The agent use ensures the system maintainability, its ability to cope with newer requirements and the possibility to integrate other actors and systems.

The architecture has been tested by implementing three transport planning models and results show that comparable results are obtained. The idea is to continue testing the architecture with diverse scheduling algorithms and negotiation schemes and with a distributed test bed.

References

1. Ambrosino, G. et al, EBusiness Applications to Flexible Transport and Mobility Services. 2001. Available online at: <http://citeseer.nj.nec.com/ambrosino01ebusiness.html>.
2. Bellifemine, F. et al, JADE - A FIPA Compliant Agent Framework. C SELT Internal Technical Report, 1999.
3. Bürckert, H; Fischer, K.; et al.. "TeleTruck: A Holonic Fleet Management System", *14th European Meeting on Cybernetics and Systems Research*, 1998, pp 695-700.
4. Coslovich, L.; Pesenti, R.; Ukovich, W. "A Two-Phase Insertion Technique of Unexpected Customers for a Dynamic Dial-a-Ride Problem". Technical Report Working paper, Universita di Trieste, Italy, 2003.
5. Cubillos, C. et al. Multi-Agent Infrastructure for Distributed Planning of Demand-Responsive Passenger Transportation Service. IEEE Int. Conf. on SMC, 2004, pp. 2013 – 2017.
6. Fischer, K.; Müller, J.P.; Pischel, M. Cooperative Transportation Scheduling: An application Domain for DAI. *Journal of Applied Artificial Intelligence*, Vol. 10, 1996.
7. Horn, M.E.T. "Fleet Scheduling and Dispatching for Demand-Responsive Passenger Services". *Transportation Research C*, Vol. 10C, 2002, pp. 35-63.
8. Jih, W; Hsu, J. Dynamic Vehicle Routing using Hybrid Genetic Algorithms. IEEE Int. Conf. on robotics & Automation. Detroit, May, 1999.
9. Kikuchi, S.; Donnelly, R.A. "Scheduling Demand-Responsive Transportation Vehicles using Fuzzy-Set Theory". *Journal of Transportation Engineering*, Vol. 118, No. 3, 1992, pp. 391-409.
10. Kohout, R; Erol, K. Robert C. In-Time Agent-Based Vehicle Routing with a Stochastic Improvement Heuristic. AAAI/IAAI Int. Conf. Orlando, Florida, 1999, pp. 864-869.
11. Miyamoto T.; Nakatyou, K.; Kumagai, S. Route planning method for a dial -a-ride problem, IEEE Int. Conf. on SMC, Vol. 4, 2003, pp. 4002 – 4007.
12. Montemanni, R.; Gambardella, et al. A new algorithm for a dynamic vehicle routing problem based on ant colony system, 2nd Int. Workshop on Freight Transportation and Logistics, 2003.
13. Perugini, D.; Lambert, D.; et al. "A distributed agent approach to global transportation scheduling". IEEE/ WIC Int. Conf. on Intelligent Agent Technology, 2003, pp 18-24.
14. Teodorovic, D.; Radivojevic, G. "A Fuzzy Logic Approach to Dynamic Dial-A-Ride Problem". *Fuzzy Sets and Systems*, Vol. 116, 2000, pp. 23-33.
15. Uchimura, K. et al. "Demand responsive services in hierarchical public transportation system". *IEEE Trans. on Vehicular Technology*, Vol. 51, Issue 4, 2002, pp. 760 – 766.
16. Weiss, G., *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, Massachusetts, USA. 1999.