New Construction Heuristics to Solve the Vehicle Routing Problem With Time Windows. New Software and Methodology

Eduardo GUILLEN SOLORZANO
Dept. Economic Analysis and Business Administration
University of La Coruña,
La Coruña, 15004, Spain

And

Manuel MARINEZ CARBALLO
Dept. Economic Analysis and Business Administration
University of La Coruña,
La Coruña, 15004, Spain

And

Susana BARBEITO ROIBAL
Dept. Economic Analysis and Business Administration
University of La Coruña,
La Coruña, 15004, Spain

ABSTRACT

In this paper we present an analysis of the traditional tour construction heuristics for the vehicle routing problem, providing new construction techniques that outperform some of the traditional methods for the benchmark problems. A wide description of the traditional methods is presented, compared to the new constructional procedures developed in the framework of this research, and based on a new combination of the addition and insertion procedures. All methods are tested on Solomon's 56 benchmark problems, for which the method developed provides better results for problems.

Keywords: VRPTW, Logistics, heuristics.

Introduction

The VRPTW problem is defined by the inclusion of time windows constraints to the Vehicle Routing Problem. In the VRP the final objective is to design minimum-cost vehicle routes in order to visit a set of customers geographically dispersed and with known demands. All customers must be serviced only once. All vehicles must depart and arrive to a central location known as depot and they cannot exceed their capacity limitations. Time windows constraints include the earliest and latest delivery times as well as the servicing time for all customers. Therefore, all the routes must consider this time windows in their stops and the vehicles must arrive earlier than the closing time of the depot.

The solution techniques for the VRPTW can be classified into three different groups. First, the construction heuristics, where a set of procedures and techniques are initially used to construct the routes that the vehicles will follow. On each route it is consider a set of customers' services attending to the priorities established by the algorithm. These methods build a solution from scratch and provide a feasible solution for making the visits with minimum cost. Secondly, we consider the improvement heuristics. These methods generally begin from an initial solution provided by a construction heuristics, and then a number of changes and node combinations are tested in order to provide a better solution. These combinations are based on 2-opt, 3opt, Or-opt, or 4-opt arc interchanges. These algorithms generally provide a better solution in the neighbourhood of solutions considered, referring to the only possibilities that the combination procedure allows (i.e. 2-opt, 3-opt, Or-opt, and 4-opt). Therefore the solution obtained is said to be *locally optimal*, but sometimes it is far from the global optimum. Finally the third group of methods are known as metaheuristics, including taboo search, simulated annealing, and genetic algorithms. The philosophy of these methods is to avoid falling into local optima by using escaping techniques. For instance taboo search uses a penalty for those movements already considered in order to explore new solutions of the neighbourhood. Therefore, the search is enriched by analysing new possible combinations. On the other hand simulated annealing considers random jumps from a solution to a different solution following a decreasing parameter called temperature. The final solution is that of the frozen state. In both methods the technique used to calculate the routes is normally based on the improvement heuristics mentioned above. Therefore the originality of these methods rely on the search strategy, but not on the construction procedure itself. On the other hand genetic algorithms consider different node combinations in order to create new generations of solutions. As if gene sequences were, the nodes are combined once and again to create new populations. The idea is to use the good node combinations obtained by any method in order to create new solutions. The philosophy underlying this method is also based on the search process, but not on the construction of the routes itself. Since it is based on a "trial and fail" approach.

In the last few years there has been an increasing number of papers dealing with the solution of the well known VRPTW, through the implementation of both improvement heuristics and metaheuristics, knowing beforehand the superiority of these methods over the construction heuristics. However very few papers deal with construction heuristics because of their bad performance in improving final results. It seems that most of the work in the last few years pursue for faster and better solutions by implementing search strategies based in exhaustive calculations for providing the final solution independently to how this solutions are built. However, if we consider that the better the initial solution is, the lower computational effort is needed by the improvement or metaheuristic algorithm to obtain a better result. Therefore we must conclude that it might be a good idea to improve the construction procedures in order to gain both in effectiveness and efficiency.

This is the motivation for which this paper deals with route construction heuristics for the Vehicle Routing Problem with Time Windows, knowing beforehand that the solutions will not beat the best solutions known.

The construction heuristic procedures are based on rational construction techniques that provide a feasible solution from the beginning. This construction techniques include: assignment procedures, addition mechanisms, and insertion heuristics, both single and double insertions are considered.

Construction Heuristics

The existing literature before the milestone paper of Marius M. Solomon, is mainly referred to case studies (Pullen and Webb, 1967; Knight and Hofer 1968 and Cook and Russell 1967), therefore the comparisons to be made before that dated were not significant. It is in 1987 when Solomon sets a set of 56 standard problems, that constitute from that date a true benchmark database, since they are adopted by the vast majority of researchers in order to test their methods. These problems are divided into six groups classified attending to the dispersion of nodes (*R* Randomized, *RC* Semi- Clustered, and *C* Clustered) and to the time horizon (Type 1, Short scheduling horizon, and type 2 Long scheduling horizon). The differences between the instances of each group stand on the percentage of customers including time window constraints, the tightness of these time windows, and the service time.[1]

From that date and on, most authors refer to the standard problems in order to test the efficiency of their methods and it is maintained an updated results database where the best results are presented.

More recent work has been done in construction heuristics, although most of it is based on Solomon methods. In 1987 Solomon develops 6 construction heuristics for the VRPTW:

These algorithms included an extension of the Clark and Wright (1964) savings heuristic adapted to a time constrained problem. In this method all customers are initially serviced by independent vehicles. From the beginning the method calculates the savings originated by the sequential visit of two customers *i* and *j* by the same vehicle as opposed to servicing them individually. The links made among the customers follow a criterion of maximising the savings created and finally routes are reported. In this case the author considered the orientation of the route due to the consideration of time. Therefore spatial and time variables were considered. Therefore when linking two customers sometimes it is necessary to compute the corresponding waiting time. The author develops a second version of the Savings algorithm but including a limitation of the waiting time, so that vehicles would not wait more than a known decision parameter.

Development of the model

The model presented in this paper is included among the construction methods in this case the construction procedure is based on a parallel construction of the routes from the beginning. The fundamentals are based on nearest neighbour additions, as well as on the insertion heuristics proposed by Solomon, but including some slight differences that improves the efficiency in some cases.

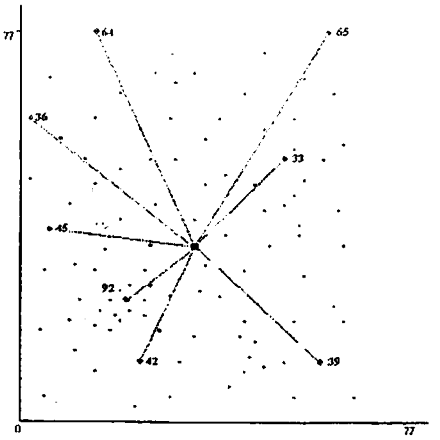
In order to illustrate the behaviour of the algorithm we have used instance R103 as an example. The notation used is the following
La notación empleada es la siguiente:

N	Total number of customers
i	Index of customers: $i = 1, 2, \dots, N$
x_i	Horizontal coordinate for customer i
y_i	Vertical coordinate for customer i
t_{ij}	Travelling time between customers i and j
ts_i	Service time at customer i.
V	Available number of vehicles
k	Vehicle index $k=1,\dots,V$
q	Vehicle capacity
d_i	Demand at customer i
ta_i	Opening time for customer i
tc_i	Closing time for customer i
ta_0	Opening time for depot
tc_0	Closing time for depot
R	Number of seed customers
β	Extra distance penalty parameter
γ	Increasing factor for the insertion limitation area for double insertions
te_{ki}	Vehicle k waiting time at customer i.
LR_k	Sequence of visits for vehicle k.
tr_k	Timing for vehicle k.
c_k	Total freight of vehicle k.
CR_k	Position of vehicle k.
D_k	Distance travelled by vehicle k.
E_k	Total waiting time for vehicle k.

The performance of the method is based on the consideration of two different stages. On the first stage it must be defined a parameter *R* referred to the number of routes that the algorithm must initially consider.

---

[1] For a more detailed description of benchmark problems, see Solomon 1987.



This number of routes will visit independently a set of R seed customers. The selection of the seed customers is made according to the urgency of their visits. The urgency is set as the time gap existing between the closing time of each node and the time needed by a vehicle to visit that node. Formally

$$h_i = tc_i - t_{0i}$$

Different vehicles will visit each of these nodes independently. However, depending on the size of this time gap, these vehicles might visit intermediate nodes on the way to the seed nodes. Therefore, for some of the nodes, the algorithm will consider insertions, and for others, the algorithm will consider direct additions. The decision is based on the size of this time gap. It is known that for Solomon benchmark problems all service times are equal, therefore, we can compare the time gap with this service time in order to know if there is time enough or not to consider intermediate visits. However the algorithm was programmed considering the possibility for different service times.

For those nodes where $h_i < ts_i$ then the vehicle has no time enough for stopping at any point, and therefore the algorithm considers a direct addition. If necessary the algorithm will calculate the corresponding waiting time before the node opens $te_{ki} = [ta_i - (t_{0i})]^+$

However, if the time gap is enough for considering insertions, then the algorithm searches all the possibilities among all the unrouted, and feasible nodes.

Feasibility is defined attending to the own restrictions of the problem. Let m be the candidate for a possible insertion, then

$$t_{0m} + ts_m + t_{mi} \leq tc_i$$
$$ta_m + ts_m + t_{mi} \leq tc_i$$
$$ta_m \leq t_{0m}$$

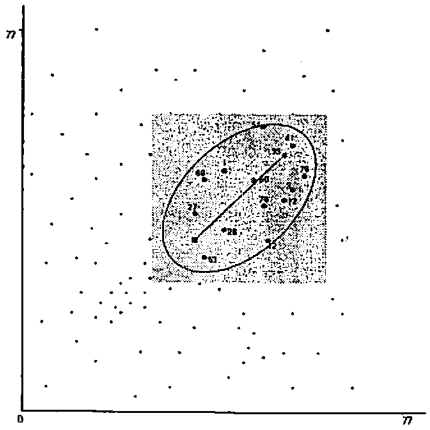
The restrictions above refer to the time needed to arrive from the depot to node m, make the service, and go from m to i. The method only considers the insertion of nodes that are open at the time the vehicle arrives at them, therefore it does not consider waiting times at nodes inserted, but it does at the final node, as it is formally indicated by the corresponding waiting time calculation $te_{ki} = [ta_i - (t_{0m} + ts_m + t_{mi})]^+$

The node insertion must also consider the constraints relative to a hypothetical return to the depot, as well as the capacity limitations of the vehicle:

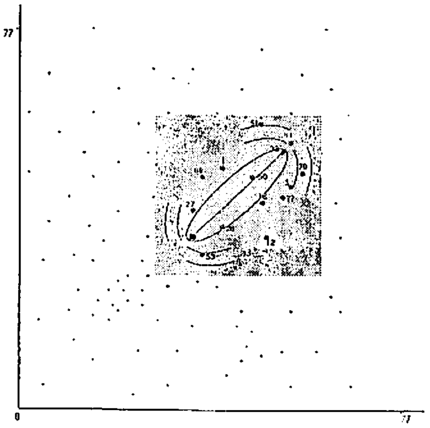
$$t_{0m} + t_{mi} + t_{i0} + te_{ki} \leq tc_0$$

$$d_m + d_i \leq q$$

All candidates to be inserted will be positioned on the ellipsoidal area denoted by the time gap of the node.



Due to the circumstances of the benchmark problems, these time gaps, can vary widely, from very tight time gaps, that do not permit the insertion of nodes, to very wide time gaps, allowing the insertion of almost all the nodes of the instance. Therefore, in the algorithm proposed, we implemented parameter for limiting the number of nodes considered for insertion. This parameter is applied to limit the extra distance permitted compared to the original distance between the depot and the destiny node. Let t_{0i} be the distance between the depot and the node, then $t_{0m} + t_{mi} \leq \beta\ t_{0i}$. Then β limits the maximum extra distance allowed for the insertions. The objective is to avoid making distant visits from the original route in order to saturate the waiting times that otherwise would appear. Therefore it can be considered as a penalty for waiting times. The smaller β is, the more penalised are waiting times are.



Despite this limitation, there are cases where there are more than one candidate for insertion, making possible multiple node insertions. In the algorithm proposed, we consider the possibility of inserting only two nodes simultaneously. If the existing time gap is wide enough, we consider pairs of nodes among the selected feasible nodes for the single insertion. In this case all possible combinations of nodes are considered. The algorithm finally selects the combination that minimises the total extra distance travelled by the vehicle. As said before, these combined visits must comply with all constraints of the problem, expressed by

$ta_m \leq t_{0m}$
$ta_l \leq t_{0m} + ts_m + t_{ml}$
$t_{0m} + ts_m + t_{ml} + ts_l + t_{li} \leq tc_i$

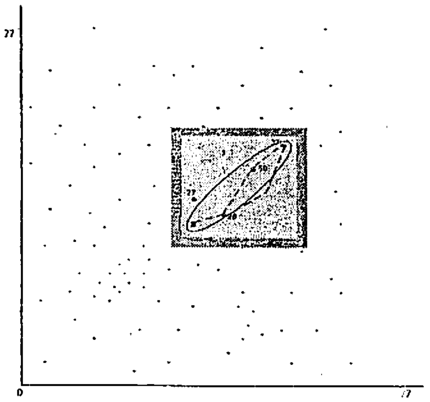
When necessary, the algorithm computes the corresponding waiting times for the last node added to the route.

$$te_{ki} = [ta_i - (t_{0m} + ts_m + t_{ml} + ts_l + t_{li})]^+$$

As before, the hypothetical return to the depot must also be on time, as well as the capacity limit constraints.

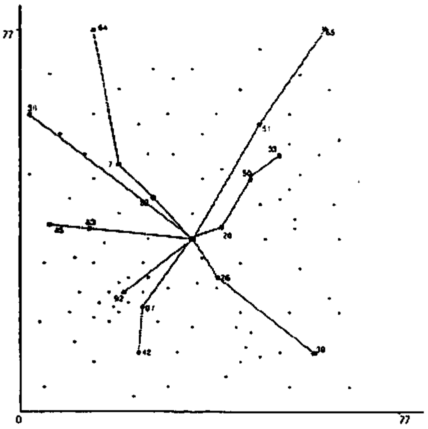
$t_{0m} + t_{ml} + t_{li} + t_{i0} + te_{ki} \leq tc_0$
$d_m + d_l + d_i \leq q$

In this cases, the combined insertions generally imply longer extra distances, although a priori they are good combinations. Therefore in order to accept this new combinations for insertion, the parameter β might be relaxed, or increased. This is implemented in the algorithm by considering a third parameter γ, that is applied as a multiplying factor to the already increased original distance. The aim is to amplify the insertion area for multiple insertions. Therefore, $t_{0m} + t_{ml} + t_{li} \leq \gamma\, \beta \cdot t_{0i}$



The final decision is based on the minimum extra distance, and when obtained, the sequence is stored in the route for the next stage.

This procedure is repeated for all the seed nodes, beginning from the nodes with lower time gaps. the reason for doing so is that the insertions considered for these nodes would be much closer to the straight line, and thus would be much better insertions than when the time gap is bigger. The resulting insertions and additions for this first stage is represented in the following chart for the problem *R103*. Right after the end of this first stage all data are updated according to the following variables: vehicle freight c_k, position of the vehicle CR_k, and route time for that vehicle tr_k. Note that at this point all vehicles have different route times, depending on the sequence of visits LR_k, time services and waiting times realized.

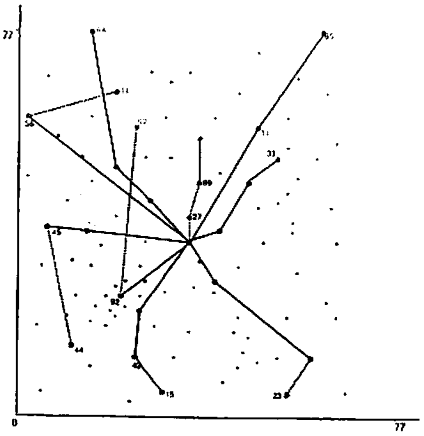


Once finished this first stage, the second and next stages begin, consisting of the additions and insertions of the rest of the nodes.

The procedures begins by the selection of the next *R* nodes according in this case to the nearest closing time of the nodes. The decision of considering the closing time instead of the time gap is based on the fact that time gaps after the first stages is a heterogeneous criterion for the vehicles since they have different timings. Therefore we must consider a homogeneous criterion as it is the closing time. The closing time indicates the urgency for the nodes, therefore the smallest closing time indicates that the node considered must be visited as soon as possible by any vehicle, since otherwise a new route should be created. Therefore the assignment of the nodes begins by those with smaller closing times.

Once the first node is selected, the algorithm analyses the needed arriving time for all vehicles to the node, and selects those vehicles that would arrive on time. For all these vehicles the one that is closer to the node is selected for making the service. In this case a nearest neighbour approach is followed by the algorithm. As in the first stage, feasibility deals with the constraints of the original problem, i.e. vehicle capacity restrictions, hypothetical arrival time to the depot, and time windows for the added node.

For every considered node, the algorithm proceeds as in the first stage analyzing all the insertion possibilities according to the intermediate nodes on the way. Both, single and multiple node insertions are considered for every selected node. However if the selected nodes cannot be assigned to any route because of the constraints of the problem, a new route is created for that node, considering again all possible insertions on the way, as it happened in the first stage. This happens in problem *R103* with the node 27, as it is shown in the following figure.



In the previous figure it can be highlighted the behaviour of the algorithm for nodes 44 and 62. for both nodes the distance from the origin to those nodes is quite big, considering that in the surroundings there were other vehicles. However, these vehicles that were in the surroundings had already departed to other nodes and thus the vehicles selected that could make the visits on time where the ones on the figure. In other cases the reason is that the only vehicle that can make the visit on time is quite far. We chose to do this instead of considering the creation of new routes.

One by one all nodes are added to the existing routes, or to new routes until there are no unrouted nodes. Once this happens all vehicles return to the depot on time, and complying with all problem restrictions, from the beginning.

Computational results

In this section awe present the results of the methods reviewed in the previous section, as they are reported by their authors.

In table 1 the corresponding average distance and average total number of vehicles used are presented for each method. All methods have been applied to Solomon's benchmark problems. Values have been rounded for simplicity.

Method	*R1*		*R2*		*RC1*		*RC2*		*C1*		*C2*	
	Dist	NV	Dist	NV	Dist	NV	Dist	NV	Dist	NV	Dist	NV
Clarke and Wright Method	1499	16,60	-	-	-	-	-	-	976	11,70	-	-
Savings, waiting time limit	1517	15,10	-	-	-	-	-	-	987	10,70	-	-
Solomon I1	1437	13,60	1402	3,30	1597	13,50	1682	3,90	952	10,00	692	3,13
Solomon I2	1639	14,50	1471	3,30	1874	14,20	1798	4,10	1050	10,10	921	3,40
Solomon I3	1652	14,10	1475	3,40	1850	14,00	1816	4,00	1103	10,00	1073	3,50
Nearest Neighbour	1600	14,50	1472	3,40	1800	14,20	1755	3,90	1171	10,20	963	3,50
Gillet and Miller Method	1500	14,60	1449	3,20	1804	14,90	1736	4,00	941	10,00	712	3,00
Potvin y Rousseau (1993)	1509	13,30	1387	3,10	1724	13,40	1651	3,60	1343	10,67	797	3.38
Ioannu et al. (2001)	1370	12,67	1310	3,09	1512	12,50	1483	3,50	865	10,00	662	3,13
Guillen et al. (2006)	1386	15,00	1095	4,00	1607	14,00	1293	5,00	924	11,00	705	3,50

Table 1

The solutions shown for the method presented in this paper were obtained after trying 4.000.000 different parameter combinations for each problem. The computational time for each combination is less than 0.01 seg, due to the determinism of the method and the limitations imposed during the construction process. Therefore it can be said that it is the fastest method in the comparison presented in this paper, even taking into account the differences due to the machines used by the different authors, which can be weighted through Dongarra's weighting factors for computational times.

The overall results indicate that the method presented outperforms many of the methods for all problems. This better performance for cases is specially good for those instances where a certain number of unilateral time windows exist. The reason for this relies on the construction process developed, which is based on a sequence of additions of nodes, inserting some unconstrained nodes in between for all the routes at the same time. Therefore the existence of these free nodes improves the results of the method. When the routes are longer, and the number of nodes included on every route is quite big, then the effect is also bigger, and the results are better.

However there are some negative effects, since during the sequential optimization procedure there are nodes that remain unvisited until the routes are almost completed. This is due to the lack of urgency of these nodes, since they probably open very soon, but close very late, therefore these nodes will not request the attention of the vehicles until the end. Therefore, at the end of the time horizon, when the routes are almost saturated sometimes it is necessary to visit these nodes and in some cases it implies the need to make new routes from the depot to visit these nodes independently. This is the reason why the number of vehicles shown in the solutions is bigger than in other methods, but it just reflects the existence of 1-node routes that could be joined, or relocated in a later improvement procedure. On the other hand the first routes to be considered, and specially the first steps of the routes report generally very good results. This is what usually happens in real time optimization when a dynamic approach is developed or even when there is some uncertainty on the definitely demands to be attended. In these cases the first nodes to be visited are combined in very good routes, but as the time runs, the visits to be done make the routes to get worse. It does not happen the same in the methods developed by Solomon, since all data are considered from the beginning in order to make the saturation of every route.

References

[1] Balakrishnan N., (1993), "Simple Heuristics for the Vehicle Routing Problem with Time Windows", *J. Opl. Res. Soc.*, Vol. 44, No. 3, p. 279-287.

[2] Clarke G. y Wright J.W., (1964), "Scheduling of Vehicles from a Depot to a number of Delivery Points", *Oper. Res.*, Vol. 12, p. 568-581.

[3] Dullaert W., (2000a), "Impact of Relative Route Length on the Choice of Time Insertion Criteria for Insertion Heuristics for the Vehicle Routing Problem with Time Windows", en *Proceedings of the Rome Jubilee 2000 Conference Improving Knowledge and Tools for Transportation and Logistics Development*: 8th Meeting of the Euro Working Group Transportation, Faculty of Engineering, "La Sapienza", University of Rome, Italy, B. Maurizio (ed), 153-156, Rome

[4] Dullaert W., (2000b), "*A Sequential Insertion Heuristic for the Vehicle Routing Problem with Time Windows with Relatively Few Custimers per Route*", Research Paper, Faculty of Applied Economics UFSIA-RUCA, 2000:014, Antwerpen, Bélgica

[5] Foisy C., y Potvin J.Y. (1993) "Implementing an Insertion Heuristic for Vehicle Routing on Parallel Hardware", *Computers and Operations Research* 20, 737-745

[6] Guillén E. (2003) Análisis de los métodos de construcción de rutas en los sistemas de planificación para el problema del VRPTW", Tesis Doctoral, Departamento de Análisis Económico y Administración de Empresas, Universidad de La Coruña, España

[7] Ioannou G., Kritikos M., y Prastacos G., (2001), "A Greedy Look-Ahead Heuristic for the Vehicle Routing Problem with Time Windows", *Journal of the Operational Research Society* 52, 523-537

[8] Potvin J.Y. y Rousseau J.M., (1993), "A Parallel Route Building Algorithm for the Vehicle Routing and Scheduling Problem with Time Windows", *Eur. J. Oper. Res.*, Vol. 66, p. 331-340.


WMSCI 2007

The 11th World Multi-Conference on
Systemics, Cybernetics and Informatics
July 8-11, 2007 – Orlando, Florida, USA.

Proceedings

IIIS
Organized by
International Institute of
Informatics and Systemics
Member of the International
Federation for Systems Research
(IFSR)

Edited by:

Agba B.
Baralt J.
Callaos N.
Chavez R-H.
El-Badawy E-S.
Fujikawa T.
Goriachkin O.
Hasan A.
Hashimoto S.

Huang S-H.
Lefevre T.
Lesso W.
Longstaff P.
Malpica F.
Pineda J.
Sanchez B.
Sanchez M.
Savoie M.

Shim E-B.
Szygenda S.
Tominaga D.
Tremante A.
Truta M.
Yang H.
Zinn C. D.
Zmazek B.

WMSCI 2007

The 11th World Multi-Conference on Systemics, Cybernetics and Informatics

July 8-11, 2007 – Orlando, Florida, USA.

PROCEEDINGS

To display the Proceedings.
insert the CD in your computer's CD-ROM drive.
Wait for the AutoRun. If AutoRun does not launch the
Proceedings, open your web browser and type the URL:
<Your CD-ROM drive letter>:\index.htm

System Requirements:
Microsoft Internet Explorer 5.0 or higher
800 x 600, High Color (16 Bit)

ISBN-10: 1-934272-13-2
ISBN-13: 978-1-934272-13-8

US $145

IIIS

Organized by
International Institute of Informatics and Systemics
Member of the International Federation for Systems Research (IFSR)

WMSCI 2007 PROCEEDINGS

IIIS

WMSCI 2007

The 11th World Multi-Conference on Systemics, Cybernetics and Informatics

July 8-11, 2007 – Orlando, Florida, USA.

PROCEEDINGS

Edited by:

Agba B.
Baralt J.
Callaos N.
Chavez R-H.
El-Badawy E-S.
Fujikawa T.
Goriachkin O.
Hasan A.
Hashimoto S.

Huang S-H.
Lefevre T.
Lesso W.
Longstaff P.
Malpica F.
Pineda J.
Sanchez B.
Sanchez M.
Savoie M.

Shim E-B.
Szygenda S.
Tominaga D.
Tremante A.
Truta M.
Yang H.
Zinn C. D.
Zmazek B.

IIIS

Organized by
International Institute of Informatics and Systemics
Member of the International Federation for Systems Research (IFSR)

VOLUME IV

CONTENTS

Bioinformatics on Post-Genomic Era - Invited Session
Organizer: Daisuke Tominaga and Sachiyo Aburatani (Japan)

Biomeasurement - Invited Session
Organizer: Shigehiro Hashimoto and Tomohiro Sahara (Japan)

Infectomics: Genomics and Mathematical/Computational Biology of Microbial Infections - Invited Session
Organizer: Sheng-He Huang and Wai-Yuan Tan (USA)

Modelling and Simulation in Mathematical and Physical Sciences - Invited Session
Organizer: Salah Doma (Egypt)

Bio-Medical Informatics and Systemics

Management Information Systems (MEI 2007)

Medical Informatics. Applications of Informatics and Cybernetics in Medicine and Health Systems

Systemics

Bio-Medical Informatics and Cybernetics

Understanding and Facing Terrorism - The Challenge of this Century - Invited Session Organizer: Meda Truta (Romania)

Management Engineering and Operations Research

Management Science

Management Support Systems and Industrial Informatics