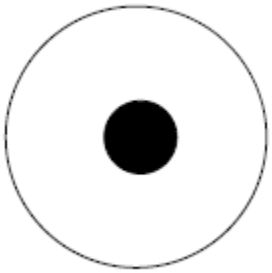
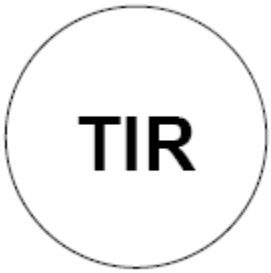




The IPSI BgD Transactions on Internet Research

Multi-, Inter-, and Trans-disciplinary Issues in Computer Science and Engineering
A publication of
IPSI Bgd Internet Research Society
New York, Frankfurt, Tokyo, Belgrade
January 2015 Volume 11 Number 1 (ISSN 1820-4503)

Special issue: "Solving Some Computationaly Hard Problems "
Guest Editor: Miodrag Zivkovic

Table of Contents:

Special Issue Articles

Solving Large Scale Instances of Hub Location Problems with a Sub-problem Using an Exact Method
Stanojević, Predrag; Marić, Miroslav……..... 1

Verification of Kurepa's Left Factorial Conjecture for Primes up to 2^{31}
Ilijaševic, Igor……..... 7

Finding Collision for MD4 Hash Algorithm Using Hybrid Algorithm
Carić, Marko…….....…….....…….....13

Permutations with Constrained Consecutive k-sums - Some Special Cases
Stefanovic, Milan; Živković, Miodrag .....…….....…….....19

Regular Articles

Cities, People, and Adapted Small Urban Space - The Geography of Real Space, Used in Real Time, by Real People
Cotner, Douglas.....…….....…….....…….....23

Do We Have Enough Information to Reduce Musculoskeletal Disorders?
Jemec, Jure; Đorđevič, Srđan…….....29

The IPSI BgD Internet Research Society

The Internet Research Society is an association of people with professional interest in the field of the Internet.
All members will receive these TRANSACTIONS upon payment of the annual Society membership fee of €500
(air mail printed matters delivery).

Member copies of Transactions are for personal use only

IPSI BGD TRANSACTIONS ON ADVANCED RESEARCH
www.internetjournals.net

STAFF		
Veljko Milutinovic, **Co-Editor-in-Chief**	**Jakob Salom,** **Co-Editor-in-Chief**	**Nemanja Trbovic,** **Journal Manager**
Department of Computer Engineering ETF University of Belgrade POB 35-54 Belgrade, Serbia Tel: (381) 64-1389281	Department of Computer Science Mathematical Institute of SANU University of Belgrade POB 367 Belgrade, Serbia Tel: (381) 64-8183030	Department of Computer Science IPSI Internet Research Society University of Belgrade POB 35-54 Belgrade, Serbia Tel: (381) 60-3191011
vm@eft.rs	**jakob.salom@yahoo.com**	**ntrbovic@gmail.com**
EDITORIAL BOARD		
Lipkovski, Aleksandar	**Gonzalez, Victor**	**Milligan, Charles**
The Faculty of Mathematics, Belgrade,	University of Oviedo, Gijon,	Sun Microsystems, Colorado
Serbia	Spain	USA
Blaisten-Barojas, Estela	**Janicic, Predrag**	**Kovacevic, Milos**
George Mason University, Fairfax, Virginia	The Faculty of Mathematics, Belgrade	School of Civil Engineering, Belgrade
USA	Serbia	Serbia
Crisp, Bob	**Jutla, Dawn**	**Neuhold, Erich**
University of Arkansas, Fayetteville, Arkansas	Sant Marry's University, Halifax	Research Studios Austria, Vienna
USA	Canada	Austria
Domenici, Andrea	**Karabeg, Dino**	**Piccardi, Massimo**
University of Pisa, Pisa	Oslo University, Oslo	Sydney University of Technology, Sydney
Italy	Norway	Australia
Flynn, Michael	**Kiong, Tan Kok**	**Radenkovic, Bozidar**
Stanford University, Palo Alto, California	National University of Singapore	Faculty of Organizational Sciences, Belgrade
USA	Singapore	Serbia
Fujii, Hironori	**Kovacevic, Branko**	**Rutledge, Chip**
Fujii Labs, M.I.T., Tokyo	School of Electrical Engineering, Belgrade	Purdue Discovery Park, Indiana
Japan	Serbia	USA
Ganascia, Jean-Luc	**Patricelli, Frederic**	**Mester, Gyula**
Paris University, Paris	ICTEK Worldwide L'Aquila	University of Szeged, Szeged
France	Italy	Hungary

Solving Large Scale Instances of Hub Location Problems with a Sub-problem Using an Exact Method

Stanojević, Predrag and Marić, Miroslav

Abstract - ***Large scale instances of NP-hard problems cannot be solved to optimality with limited computational resources. A general approach is to use a heuristic method to obtain an approximate solution. However, when the model of a problem contains a sub-problem, it is possible to perform an exhaustive search on some subset of all sub-problems, thus improving solutions or confirming the optimality of existing solutions on the given subset. In this paper we demonstrate one such approach, for solving the Single Allocation Hub Location Problems (SAHLP), the capacitated version (CSAHLP) and the uncapacitated (USAHLP) version. An exhaustive search is performed for all hub configurations with small number of hubs (the location part) and for each one CPLEX is employed to solve the sub-problem for that hub configuration (the allocation part of the whole problem). We tested this method on USAHLP and CSAHLP test instances between 100 and 200 nodes, for which the optimal solutions are not known. Our extensive computational tests have resulted in 2 improved solutions and confirmed that other best known solutions may be considered optimal, in the sense of tested hub configurations with a certain number of hubs.***

Index Terms - hub location, sub model, discrete optimization, exhaustive search

1. INTRODUCTION

HUB location problems consist of two parts: the location part establishes hubs at some nodes in a given network and the allocation part determines the allocation of non-hub nodes to newly established hubs. The objective function and the constraints can vary between different types of hub location problems. In this study we focus on minimization of a combination of the total transportation cost and the fixed cost of establishing hubs. For the constraints, we consider single allocation type, which means that a non-hub node is allocated to only one hub. If each hub has a predetermined capacity, the problem is known as the Capacitated Single Allocation Hub Location Problem (CSAHLP). If hubs have unlimited capacities, we are dealing with the Uncapacitated Single Allocation Hub Location Problem. USAHLP has been studied extensively and solved either using exact or heuristic approaches [1], [2], [3], [4]. More recent papers on USAHLP are [5], [6], [7]. An overview of papers dealing with CSAHLP is given in [8]. Different formulations of the problem are given in [9], [10], [11] and [12] and the most successful heuristic methods for solving CSAHLP are presented in [11], [13], [14] and [15].

The method presented herein was inspired by our research which resulted in paper [17].

Manuscript received January 13, 2014. This research was partially supported by Serbian Ministry of Science and Technological Development under the grant no. 174010.

P. Stanojević is a PhD candidate at the Faculty of Mathematics, University of Belgrade, Serbia (e-mail: djape@sbb.rs)

M. Marić is an assistant professor at the Faculty of Mathematics, University of Belgrade, Serbia (e-mail: maricm@matf.bg.ac.rs).

2. MATHEMATICAL MODELS

2.1 Full model

Correia et al. demonstrated in [10] that an additional set of constraints must be added to the mathematical model of CSAHLP in order to avoid potentially infeasible solutions. Therefore, we use this revised formulation of the CSAHLP from [10], including the missing cuts. The same model was used in [17].

A set of n distinct nodes represents a network I. Every two nodes in the network are connected and two matrices are given, W_{ij} and C_{ij}.

W_{ij} is the flow matrix, it represents the amount of flow from an origin i to destination j. This matrix is not necessarily symmetric, and W_{ii} may be greater than 0.

The matrix C_{ij} represents transportation cost per unit flow between any two nodes i and j from the network I. The transportation cost C_{ij} could be any abstract value corresponding to an ordered pair of two nodes i *and* j. In practice, however, the value C_{ij} often depends on the distance between i and j and for this reason, for two nodes l *and* k, $l \neq k$, we will often say that l *is the node nearest to node* k which technically means that for a given k, the value C_{kl} is the lowest.

In CSAHLP, each potential hub has limited capacity. This collection capacity of each node k is given by G_k.

Establishing a hub incurs some cost, which will be a part of the objective function. The cost of establishing a hub differs between locations and therefore, for every node *k* a fixed cost value is given by a constant F_k. In literature on USAHLP it has been noted that this fixed cost greatly affects the objective function. For this reason, each test instance of USAHLP is usually tested with three different values of the f_k vector: $f_k = F_k$, $f_k = 0.9*F_k$ and $f_k = 0.8*F_k$. For CSAHLP we will only use $f_k = F_k$.

Standard formulation of xSAHLP (CSAHLP or USAHLP) problems states that direct transportation between non hub nodes is not allowed, it must be routed via hubs. This means that any flow between an origin *i* and a destination *j* is routed via some hub nodes *k* and *l* respectively, where $k,l \in H$ and $H \subseteq I$ is a set of established hubs and node *i* is always allocated to hub *k* and node *j* is always allocated to hub *l* (single allocation). Hence, the total cost of transportation between *i* and j consists of three parts: the costs of collection (from origin *i* to hub *k*), transfer (from hub *k* to hub *l*) and distribution (from hub *l* to destination *j*). Collection, transfer and distribution should each incur different weighted costs and this is given by the parameters χ, α and δ, respectively. The transfer between the hubs should have lower cost than collection and distribution, so it is assumed that χ,δ>α. Therefore, the total transportation cost per unit flow from an origin *i* to a destination *j*, via hubs *k* and *l*, is equal to $\chi C_{ik}+\alpha C_{kl}+\delta C_{lj}$.

Finally, we introduce binary decision variables Z_{ij}, $i,j \in I$, which determine both the location of hubs and the allocation of non-hubs to hubs. Assuming $H \subseteq I$ is a set of established hubs, then $Z_{kk}=1 \Leftrightarrow k \in H$, so hubs are always allocated to themselves. Also, Z_{ij} takes the value of 1 if node *i* is allocated to a hub node $j \in H$, and 0 otherwise. Since no direct flow between non-hub nodes *i* and *j* is allowed, continuous non-negative decision variables are added to the model: Y_{kl}^{i} represent the amount of flow originated from node *i*, collected at hub *k* and distributed via hub *l*. The triangle-inequality holds for the transportation costs between nodes, so the flow will always be routed via at most two hubs. Finally, O_i and D_j represent the amount of flow which departs from $i \in I$ and the amount of flow that is distributed to node $j \in I$, respectively, i.e. $O_i=\sum_{j\in I} W_{ij}$ and $D_j=\sum_{i\in I} W_{ij}$.

Using this notation, we formulate one model for xSAHLP, which is used both for the CSAHLP and USAHLP, with only one set of constraints that is different between the two problems:

$$\textbf{minimize} \sum_{i\in I}\sum_{k\in I} C_{ik}Z_{ik}(\chi O_i + \delta D_i) + \sum_{i\in I}\sum_{k\in I}\sum_{l\in I} \alpha C_{kl} Y_{kl}^{\,i} + \sum_{k\in I} f_k Z_{kk} \quad (1)$$

subject to:

$$\sum_{k\in I} Z_{ik} = 1 \quad \text{for every } i \in I \quad (2)$$

$$Z_{ik} \le Z_{kk} \text{ for every } i,k \in I \quad (3)$$

$$\sum_{l\in I} Y_{kl}^{\,i} - \sum_{l\in I} Y_{lk}^{\,i} = O_i Z_{ik} - \sum_{j\in I} W_{ij} Z_{jk} \text{ for every } i,k \in I \quad (4)$$

$$\sum_{i\in I} O_i Z_{ik} \le G_k Z_{kk} \quad \text{for every } k \in I \quad (5^*)$$

$$\sum_{l\in I, l\neq k} Y_{kl}^{\,i} \le O_i Z_{ik} \text{ for every } i,k \in I \quad (6)$$

$$Y_{kl}^{i} \ge 0 \text{ for every } i,k,l \in I \quad (7)$$

$$Z_{ik} \in \{0,1\} \text{ for every } i,k \in I \quad (8)$$

CSAHLP uses all of the above constraints; USAHLP does not include the constraint set (5), which states that the collection in each hub must be limited by hub's capacity.

The first term in the objective function (1) is the sum of all collection and delivery costs; the second term is the sum of all transfer costs and the third term is the sum of fixed costs for establishing hubs. The constraint set (2) ensures single allocation: each node is allocated to exactly one other node and the constraints set (3) enforces that if any node is allocated to a node *k* then *k* must also be allocated to *k*, i.e. *k* is a hub. The constraint set (4) enforces the flow conservation equality in the network. The constraint set (6) ensures that if *i* is not allocated to *k,* then there is no flow from *i* through *k* to any destination *l*. This is the missing cuts set introduced in [10]. Variables Y_{kl}^{i} are continuous and non-negative (7), while the variables Z_{ij} are binary (8).

The model with all constraint sets will also be referred to as CSAHLP and the model without (5) will be referred to as USAHLP, while xSAHLP will be used to refer to both models.

2.2 Model for a fixed set of hubs

Using the same notation as in section 2.1, let us assume that a set of hubs $H \subseteq I$ is given, i.e. it is predetermined and fixed. The location part in that case is done, but the allocation of non-hubs to hubs still needs to be optimized. For a given set

of hubs $H \subseteq I$, we are dealing with a sub-problem of the xSAHLP, denoted as xSAHLP(H). In [16] O'Kelly demonstrates that even this problem is NP hard.

For a given $H \subseteq I$, the sum of costs of establishing hubs is determined, so we need an objective function which minimizes the sum of all transportation costs and the constraints which accommodate for this change:

$$\textbf{minimize} \sum_{i\in I\setminus H}\sum_{k\in H} C_{ik}Z_{ik}(\chi O_i + \delta D_i) + \sum_{k\in H} C_{kk}(\chi O_k + \delta D_k) + \sum_{i\in I\setminus H}\sum_{k\in H}\sum_{l\in H} \alpha C_{kl}{Y_{kl}}^i + \sum_{i\in I\setminus H}\sum_{k\in H}\sum_{l\in H} \alpha C_{lk}W_{li}Z_{ik} + \sum_{k\in H}\sum_{l\in H} \alpha C_{kl}W_{kl} + \sum_{k\in H} F_k \quad (9)$$

$$\textbf{subject to:}$$

$$\sum_{k\in H} Z_{ik} = 1 \quad \text{for every } i \in I\setminus H \quad (10)$$

$$\sum_{l\in H} {Y_{kl}}^i - \sum_{l\in H} {Y_{lk}}^i = (O_i - W_{ii})Z_{ik} - W_{ik} - \sum_{j\in I\setminus H, j\neq i} W_{ij}Z_{jk} \quad \text{for every } i \in I\setminus H, k \in H \quad (11)$$

$$O_k + \sum_{i\in I\setminus H} O_i Z_{ik} \leq G_k \quad \text{for every } k \in H \quad (12^*)$$

$$\sum_{l\in H, l\neq k} {Y_{kl}}^i \leq O_i Z_{ik} \text{ for every } i \in I\setminus H, k \in H \quad (13)$$

$$Y_{kl}^i \geq 0 \text{ for every } i \in I\setminus H, k, l \in H \quad (14)$$

$$Z_{ik} \in \{0,1\} \text{ for every } i \in I\setminus H, k \in H \quad (15)$$

Again, the constraint set (12), which corresponds to the constraint set (5) of the full model, is used only for CSAHLP; hence, CSAHLP(H) is this model with the constraint set (12) and USAHLP(H) is the same model but without (12).

Since each hub from $H \subseteq I$ must be allocated to itself, and having in mind that variables Z_{ik} are binary, the constraint set (3) has become trivial and can be removed from both xSAHLP(H) models. All other constraint sets correspond to the same constraint sets from the full model. The difference is that for Z_{ik} variables, the first index i takes values from the set $I\setminus H$ only, and the second index k takes values from the set H (15). Likewise, for the ${Y_{kl}}^i$ variables, $i \in I\setminus H$ and $k,l \in H$ (14). Since the hubs are already known, the objective function is reformulated to define the minimization of the sum of transportation costs (9).

2.3 Test approach

If $h=|H| << |I|=n$, the number of variables and the number of constraints in the xSAHLP(H) are significantly lower compared to xSAHLP model. Instead of n^3+n^2 variables in the xSAHLP models, there are $(n-h)*h^2+(n-h)*h$ variables in the xSAHLP(H). The case of $|H| << |I|$ usually occurs in most situations from practice. Namely, for Australia Post test instances between 100 and 200 nodes, obtained from the Australian Post network [18], optimal solutions are not known, but the number of hubs in the best known solutions is either 2 or 3 or 4. When we attempted to solve the full CSAHLP with 100 nodes and USAHLP with 120 nodes using CPLEX, the solver ran out of memory, because it was dealing with millions of variables and constraints. However, for a given H with 2 or 3 or 4 hubs, xSAHLP(H) would only have to fix between 600 and 2000 variables. Of course, the number of possible hub configurations with $h=2,3,4$ grows rapidly, but with sufficient computational resources it was plausible to attempt to solve them all. Choosing h hubs from n nodes can be done in $\binom{n}{h}$ different ways.

For AP test instances, the number of these combinations is as follows:

$\binom{100}{2} = 4950$ $\quad$ $\binom{130}{2} = 8385$

$\binom{100}{3} = 161700$ $\quad$ $\binom{130}{3} = 357760$

$\binom{100}{4} = 3921225$ $\quad$ $\binom{130}{4} = 11358880$

$\binom{120}{2} = 7140$ $\quad$ $\binom{200}{2} = 19900$

$\binom{120}{3} = 280840$ $\quad$ $\binom{200}{3} = 1313400$

$\binom{120}{4} = 8214570$ $\quad$ $\binom{200}{4} = 64684950$

Finally, in order to improve the best known solutions we do not need to find the exact solution of xSAHLP(H) for each H; we are only looking for a solution that is better than the currently best known solution. Therefore, if *upperbound* is the value of the best known solution of a particular test instance, for any given H we use *upperbound* as the upper bound parameter in CPLEX (denoted CPLEX(xSAHLP(H), upperbound)), so that it can stop optimizing if a solution better than that value cannot be found. This significantly decreases computational time, because, for a given H, it often happens that either just the fixed costs of establishing hubs are greater than the *upperbound,* so there is no need to invoke CPLEX, or CPLEX quits immediately or after traversing only a few nodes because the solution of the LP relaxed problem for the given H is worse than the *upperbound.*

The following algorithm illustrates our approach:

```
Algorithm: xSAHLP
input : matrices as in section 2.1
input : upper bound b
input : number of hubs h
output: solutions better than b

for each H⊆I, |H|=h do
  if (fixedCosts(H)<b)
    solution = CPLEX(xSAHLP(H),b)
    if (solution exists) then
      output solution
      update b
    end if
  end if
end for each
end algorithm xSAHLP
```

3. COMPUTATIONAL RESULTS

All tests were performed on three identical hardware configurations with Intel Core i7-860 2.8 GHz processors and 8GB RAM memory under Windows 7 Professional operating system. CPLEX 12.1 package was used for solving xSAHLP(H) and xSAHLP algorithm was implemented in C programming language. The tests were performed on AP test instances with between 100 and 200 nodes for which the optimal solutions are not known. The best known solutions for USAHLP were taken from [6] and the best known solutions for CSAHLP were taken from [17] and [13].

The results for USAHLP are shown in Table 1 and the results for CSAHLP are shown in Table 2. The columns in Table 1 are as follows: the name of the test instance (the same name as in [6]); the fixed cost factor (either 1.0 or 0.9 or 0.8, as discussed in section (2.1)); the best known result from literature up to now; the upper bound value used for testing, which is equal to the best known value, rounded up to the nearest integer; and finally the results obtained by our tests and the test time in seconds, for h=2,3,4, including the total time in days for each test instance. The columns in Table 2 are the same as in Table 1 except that there is no fixed cost factor. The test instance names in Table 2 are the same as in [17].

Two results better than the best known solution have been obtained, both for the USAHLP test instance 200T with 0.9 and 0.8 fixed cost factors. For all other test instances in Table 1, the results obtained are the same as the best known solution. However, for all USAHLP test instances tests were also performed for all hub configurations with one hub more than in the currently best known solution. Therefore, this gives a strong suggestion that the currently best known solution is actually optimal. As the number of hubs increases, the fixed cost factor significantly outweighs the transportation costs in the objective function and therefore, better solutions cannot be obtained. This is also visible from the fact that the test time was longer with a lower fixed cost factor than with a higher fixed cost factor, for the same test instance. Total computational time for all USAHLP tests was 40.38 days.

All best known solutions for CSAHLP test instances with 100 nodes were found and also a test with one more hub than in the best known solution was performed. Two additional tests for the test instance 100TT were also performed but are not shown in the table; first with h=5 and it took 37691 seconds to test all $\binom{100}{5}=75287520$ hub configurations; second with h=6, meaning that $\binom{100}{6}=1192052400$ (≅1.2billion) hub configurations were tested and in just 54861 seconds. This short running time (comparative to the number of tested hub configurations) means that the fixed cost of establishing 5 or 6 hubs was greater than the best known solution, for almost all 5-hub and 6-hub configurations. It also means that it is safe to say that 474068.96 is the optimal solution for CSAHLP test instance hub100TT.

CSAHLP test results for test instances with 200 nodes are not so conclusive. The best known solutions for hub200LT and hub200TT have 4 hubs and testing 64 million 4-hub configurations was not feasible. It took about 7 days to test all 3-hub configurations. For the test instance 200TL the best known solution has 3 hubs and it was confirmed as the best solution among all 3-hub configurations in less than one day.

CSAHLP test instance 200LL requires special attention. The best known solution for this test instance, published by Contreras et al. in [13] is 231069.50 is significantly better than the best known solution published in other papers ([11], [14], [17]), which was 241992.97. Our tests demonstrate that 241992.97 is the best solution from all 2-hub configurations and that there is no solution better than that with 3 hubs. The best solution with 4 or more hubs obtained by the evolutionary algorithm in [17] is 251369.64. To verify the solution published in [13] we contacted the authors of that paper, but to this day after more than 6 months of waiting, we have not received a response from the authors. Any given hub configuration H with any number of hubs can easily be tested and optimized using the CSAHLP(H) model.

4. CONCLUSION

The approach used herein has improved 2 best known solutions up to now and confirmed that other best known solutions are optimal for all hub configurations with a given number of hubs (2, 3 or 4). It also demonstrates that the standard test instances which are widely used in literature do not reflect the theoretical complexity of the mathematical model xSAHLP. Test instances with lower fixed costs are needed, as they allow for more hubs to be established and thus increase the complexity. The value of $\binom{n}{k}$ is the largest for k=n/2, however, the fixed cost of

establishing any k=n/2 hubs would by far surpass the existing best known solutions for all available test instances.

Furthermore, the same approach can be used for any problem with a sub problem. If in a model M the set of variables is V=A∪B and if fixing all variables from the set A results in a sub-model M(A) with variables B, then the model M(A) can be tested for some configurations of the set A. Depending on which configurations are chosen for testing, this could lead to improved results.

REFERENCES

[1] Abdinnour-Helm S, "A hybrid heuristic for the uncapacitated hub location problem", European Journal of Operational Research 1998; 106: 489–499.

[2] Abdinnour-Helm S, Venkataramanan MA, "Solution Approaches to Hub Location Problems", Annals of Operations Research 1998; 78: 31–50.

[3] Skorin-Kapov D, Skorin-Kapov J. "On tabu search for the location of interacting hub facilities", European Journal of Operations Research 1994; 73: 502–509.

[4] Topcuoglu H, Court F, Ermis M, Yilmaz G. "Solving the uncapacitated hub location problem using genetic algorithms." Computers & Operations Research, 2005; 32: 967–984

[5] Filipović V, Kratica J, Tošić D, Dugošija Dj. "GA Inspired Heuristic for Uncapacitated Single Allocation Hub Location Problem". Advances in Soft Computing 2009; 58: 149–158.

[6] Marić M, Stanimirović Z, Stanojević P. "An efficient memetic algorithm for the uncapacitated single allocation hub location problem", Soft Computing, 2013; 17(3): 445–466.

[7] Silva MR, Cunha CB. "New simple and efficient heuristics for the uncapacitated single allocation hub location problem", Computers & Operations Research, 2009; 36: 3152–3165.

[8] Alumur S, Kara BY. "Network hub location problems: the state of the art", European Journal of Operational Research 2008; 190(1): 1–21.

[9] Campbell JF. "Integer programming formulations of discrete hub location problems". European Journal of Operational Research 1994; 72: 387–405.

[10] Correia I, Nickel S, Saldanha-da-Gama, F. "The capacitated single-allocation hub location problem revisited: A note on a classical formulation." European Journal of Operational Research, 2010; 20: 792–796.

[11] Ernst AT, Krishnamoorthy M. "Solution algorithms for the capacitated single allocation hub location problem". Annals of Operations Research 1999; 86: 141–159.

[12] Skorin-Kapov D, Skorin-Kapov J, OKelly M. "Tight linear programming relaxations of uncapacitated p-hub median problems". European Journal of Operational Research 1996; 94: 582–593.

[13] Contreras I, Dias J, Fernandez E. "Lagrangean relaxation for the capacitated hub location problem with single assignment", OR Spectrum 2009; 31: 483–505.

[14] Stanimirović Z. "Solving the Capacitated Single Allocation Hub Location Problem Using Genetic Algorithm". C.H. Skiadas. ed. Recent Advances in Stochastic Modelling and Data Analysis.World Scientific Publishing Co Pte Ltd. 2007; 464–471.

[15] Randall M. "Solution approaches for the capacitated single allocation hub location problem using ant colony optimisation", Computational Optimization and Applications 2008; 39: 239–261.

[16] O'Kelly, ME. "A quadratic integer program for the location of interacting hub facilities". European Journal of Operational Research 1987; 32: 393–404.

[17] Stanojević, P, Marić M, Stanimirović Z. "A hybridization of an evolutionary algorithm and a parallel branch and bound for solving the capacitated single allocation hub location problem", Preprint submitted to Applied Soft Computing September 26, 2013

[18] Beasley JE. "Obtaining Test Problems via Internet." Journal of Global Optimization, 1996; 8(4): 429–433.

Table 1: Test results for USAHLP test instances

				2 hubs		3 hubs		4 hubs		
Test Instance	Fixed cost factor	Best known result	Upper bound	Result	Time (seconds)	Result	Time (seconds)	Result	Time (seconds)	Total time (in days)
120T	1.0	232460.82	232461	232460.82 HUBS: 31 84	150.29		3060.82		49936.76	0.62
120T	0.9	227687.23	227688	227687.23 HUBS: 31 84	166.23		3596.40		66194.45	0.81
120T	0.8	222680.29	222681		190.67	222680.29 HUBS: 31 76 84	4360.16		86464.91	1.05
120L	1.0	225801.36	225802	225801.36 HUBS: 31 84	393.82		22457.75		123898.01	1.70
120L	0.9	221693.72	221694	221693.72 HUBS: 31 84	507.41		29385.04		93385.21	1.43
120L	0.8	215865.55	215866		398.80	215865.55 HUBS: 31 76 84	28677.32		918814.68	10.97
130L	1.0	227884.63	227885	227884.63 HUBS: 35 87	683.33		43548.30		*not tested*	0.51
130L	0.9	223280.66	223281	223280.66 HUBS: 35 87	750.22		49476.04		*not tested*	0.58
130L	0.8	218676.70	218677	218676.70 HUBS: 35 87	695.61		46090.87		*not tested*	0.54
130T	1.0	234935.97	234936	234935.97 HUBS: 35 87	271.39		5885.86		67712.50	0.85
130T	0.9	229626.87	229627	229626.87 HUBS: 35 87	330.38		7831.36		115687.51	1.43
130T	0.8	224317.78	224318	224317.78 HUBS: 35 87	359.61		9546.23		*137987.93*	1.71
200L	1.0	233802.98	233803	233802.98 HUBS: 42 147	4061.20		439452.67		*not tested*	5.13
200L	0.9	228753.70	228754	228753.70 HUBS: 42 147	4393.91		443346.29		*not tested*	5.18
200L	0.8	223704.42	223705	223704.42 HUBS: 42 147	5013.87		448611.59		*not tested*	5.25
200T	1.0	272188.11	272189	272188.11 HUBS: 53 121	1561.27		53131.35		*not tested*	0.63
200T	0.9	266134.11	266135		1914.48	**266116.32 HUBS: 53 94 185**	79338.66		*not tested*	0.94
200T	0.8	258823.13	258824		2137.24	**258797.48 HUBS: 53 94 185**	86997.01		*not tested*	1.03

Table 2: Test results for CSAHLP test instances

			2 hubs		3 hubs		4 hubs		
Test Instance	Best known result	Upper bound	Result	Time (seconds)	Result	Time (seconds)	Result	Time (seconds)	Total time (in days)
hub100LL	246713.97	246714		127.36	246713.97 HUBS: 28 63 72	13750.84		534551.34	6.35
hub100LT	256155.33	256156		90.67	256155.33 HUBS: 28 67 75	15252.42		581240.65	6.90
hub100TL	362950.09	362951	362950.09 HUBS: 43 51	51.11		1094.25		3449.29	0.05
hub100TT	474068.96	474069		56.00		2980.40	474068.96 HUBS: 4 33 85 94	17689.93	0.24
hub200LL	231069.50*	241993*	241992.97 HUBS: 42 158	3124.18		628788.31		*not tested*	7.31
hub200LT	267236.99	267237		2007.52		637303.66		*not tested*	7.40
hub200TT	290743.23	290744		759.08		81858.15		*not tested*	0.96
hub200TL	273443.81	273444		1247.16	273443.81 HUBS: 53 94 185	71281.27		*not tested*	0.84

Verification of Kurepa's Left Factorial Conjecture for Primes up to 2^{31}

Ilijašević, Igor

***Abstract*—In this article we utilize Graphics Processing Units (GPUs) as a way to expedite massive computation needed to check Kurepa's hypothesis on left factorials for primes up to 2^{31}. We also give an estimate of computational efforts and an existing algorithm which can be used to extend the computations up to 2^{32}.**

***Index Terms*—CUDA, GPU, Kurepa, left factorial, primes.**

I. Introduction

KUREPA [1] defined left factorial as a sum

$$!n = \sum_{i=0}^{n-1} i!, \quad n \text{ integer} \tag{1}$$

and stated the conjecture (see [2, Problem B44] and the overview paper [3]) that

$$!n \not\equiv 0 \pmod{n}, \quad n > 2, \tag{2}$$

or equivalently $\gcd(!n, n!) = 2$. Let $r_p = !p \pmod{p}$ for an arbitrary prime p. An equivalent formulation of the conjecture is

$$r_p \neq 0 \quad \text{for all primes } p > 2. \tag{3}$$

The conjecture is previously verified by Mijajlović [4] for $p \leq 311009$, Živković [6] for $p < 2^{23}$, Gallot [7] for $p < 2^{26}$, Jobling [8] for $p < 144000000$ and Tatarević [9] for $p < 10^9$.

We extended the computation up to $p < 2^{31}$ using NVIDIA CUDA GTX 285 and GTX 780 GPUs. Given p, the value of r_p (3) is computed by obvious algorithm

Algorithm 1 Calculating left factorial modulus

```
1: procedure LeftFactorialMod(p)
2:     f = 1                                  ▷ 0! = 1
3:     s = 1                                  ▷ !0 = 1
4:     for i = 1 to p do
5:         f = f * i (mod p)                  ▷ f = i! (mod p)
6:         s = s + f (mod p)                  ▷ s = !i (mod p)
7:     end for
8:     return s                               ▷ s = !p (mod p)
9: end procedure
```

Given that $p < 2^{32}$, the values of all variables can be stored in a 64-bit representation. The calculation is not straightforward, because of the lack of efficient implementation of modular arithmetic on GPU.

Manuscript received January 27, 2014.

Ilijašević Igor is a PhD student with the Faculty of Mathematics, University of Belgrade, Belgrade, Serbia (e-mail: ilijasevic.igor@gmail.com).

Milovanović [5] introduced a generalization of left factorials, the numbers

$$K_m(n) = \sum_{i=0}^{n-1} \frac{(-1)^i}{i!} \sum_{v=i}^{n-1} v! \binom{m+n}{v+m+1}, \; K_m(0) = 0. \tag{4}$$

In terms of these numbers $n! = K_{-1}(n+1)$ and $!n = K_0(n)$. Let

$$A = \{13k + 9 | k \in \mathbb{N}\}$$

Milovanović formulated two new conjectures,

$$(\forall n \in \mathbb{N}\backslash\{5\}) \quad \gcd(K_0(n), K_1(n)) = 1, \tag{5}$$

($\gcd(K_0(5), K_1(5)) = \gcd(34, 51) = 17$) and

$$\gcd(K_1(n), K_2(n)) = \begin{cases} 1, & n \in \mathbb{N}\backslash A \\ 13, & n \in A \end{cases} \tag{6}$$

After a not too hard computer search we found counterexamples to these conjectures.

II. Overview of the Graphics Processing Units and CUDA framework

In order to extend the previous range of primes for which (3) is checked, the obvious choice is to do computations in parallel.

We decided to use nVidia hardware and framework. It should be noted that while a heterogeneous programming standard OpenCL exists and is supported by both nVidia and other vendors, due to the fact that we had access to nVidia hardware and the fact that nVidia OpenCL support is both behind in terms of performance and version support, we have chosen to use nVidia only supported options. However it is possible to modify programs to comply with OpenCL.

While evaluating different GPU architectures and CUDA versions one should be aware of their respective capabilities. Performance capabilities of various GPUs vary even in the same architecture class, for example GTX 780's 1/24 compared to GTX Titan's 1/3 double precision performance with respect to single precision floating point operations. Differences are even bigger for different architectures. For example, pre-Fermi class GPUs integer operations are internally realized using 24-bit arithmetic, compared to 32-bit arithmetic in latter devices; pre-Tesla cards lack the double precision arithmetic, etc.

Our work was done primarily on GTX 285 with GTX 780 being available later merely as an additional device and as such no advanced Kepler capabilities have been utilized. Basic differences can be seen in Table I. An explanation of the

capabilities of devices with compute capability 1.3 and 3.5 can be found in [11, Chapter 2.5,Chapter 5.4.1,Appendix G.1.].

Here we present CUDA terminology used in our work.

Each CUDA capable device consists of several Streaming Multiprocessors (SM) which can vary in number.

- Each SM consists of a number of scalar in-order processors each executing same instruction at the same time using threads.
- Group of 32 threads is called a warp and is the minimum size of the data processed synchronously by SM. A warp executes one common instruction at a time, so full efficiency is realized when all 32 threads of a warp agree on their execution path.
- Blocks present a group of threads which are defined by the programmers and variable in size, all threads in a block can coordinate with each other.
- Grid is a collection of blocks which are completely independent so there is no coordination and synchronization within a grid.

There are several memory hierarchies available in CUDA:

- Global memory is SDRAM memory available on the GPU. Each thread can read and write to this memory. It is the largest and the slowest memory available on the GPU.
- Texture cache is memory within each SM that can be used as read only cache. This memory is used if there exists a high degree of spatial locality in memory accesses.
- Constant memory is read only memory available for each SM.
- Shared memory is a small amount of memory available for each SM which can be both written and read by any thread in a block running on the same SM.
- Each SM contains a number of registers available for use by CUDA program threads which are shared resources allocated among the thread blocks executing on an SM.

Occupancy is the ratio of active warps to the maximum number of warps supported on an SM of the GPU. It is calculated in the form of

$$Occupancy \ = \ \frac{Active \ Warps}{Maximum \ Active \ Warps}.$$

Increase in usage of the following resources has a negative effect on occupancy:

- Register usage
- Shared memory usage
- Block size

We shall note the size of a block as SB and the number of threads per block as TPB.

III. Implementation of modular multiplication

Integer operations on GPU are very constrained. GPUs with Tesla architecture support only 24-bit integer multiplication natively. The following generations support 32-bit integer arithmetic fully in hardware.

Modulus operation on the GPU is extremely expensive. However if the modulus is known at compile time or if it is a constant, as in $\mathbf{LeftfactorialMod}(p)$, the remainder operation can be replaced using various methods, e.g. multiplication by precomputed divisor reciprocal. We have done our calculations using three such replacements, both for Tesla and Kepler architectures. All of the following algorithms use a precomputed constant (magic number) and seek to replace the expensive modulus operations with more suitable multiply, add and shift integer operations, or double precision floating point operations. Functions which are executed on the GPUs are called kernels.

TABLE I
Basic device information; for a more detailed device specification see [12], [13].

	GTX 285	GTX 780
Architecture	Tesla	Kepler
Compute capability	1.3	3.5
Streaming Multiprocessors (SM)	30	12
Maximum number of threads per block	512	1024

It should be noted that the following algorithms utilize the **umulhi** operation which gives us the 32 most significant bits (MSB) when multiplying two unsigned 32-bit numbers. We will note the least significant bits as LSB.

As the largest possible factorial we can keep in 64 bits is 20! and given that the modulus is ever changing, we have decided to run each calculation independently in parallel preventing us from using shared variables over the entire block. In simplest terms each thread is independent and calculates factorial and left factorial for a specific prime.

As we can see, in order to achieve the best results we have to take several things in consideration:

- We should strive for maximum occupancy which forces us to take special care of number of registers used by each thread which in turn forces us to seek more "lightweight" algorithms.
- Thread divergence is unfortunately very difficult to deal with the way we have decided to tackle the problem. As the distance between primes is not constant (minimum distance = 2, average = 22, maximum = 292) one of the possible ways to solve this problem is to group primes with similar distances together. Then after finishing the calculation for the first prime in a block reshuffle the prime execution on the threads and execute them in the same warps if possible. However this method requires the use of additional shared memory and in our tests it has not shown any additional performance improvements. Given that we operate with numbers over 10^9, we can see that in average case the divergent path, while considerable, still represents a small percentage of the whole execution:
$$\frac{p}{TPB * 22}, \quad 2^{31} > p > 10^9$$
- We should avoid using integer arithmetic on the GTX 285 and 64-bit integer arithmetic on the GTX 780. Unfortunately the double precision of the cards we have available rather limited compared to nVidia's commercial scientific cards, and as such results presented here are not representative of other cards.

When writing the code we have used the tools made available by nVidia's CUDA Toolkit [10]. Some of the tools we used:

- Nsight monitor is a debugging tool invaluable in debugging the code and in bug detection regarding memory accesses.
- CUDA GPU Occupancy Calculator is used to determine the ideal theoretical block size as it contains the information regarding the various compute capabilities. It does so by giving us the theoretical occupancy of a kernel based on the number of parameters. Occupancy is given in percentages and it gives us an idea of the GPU pipeline utilization.
- Visual Profiler is a profiler valuable in showing us the various usages regarding the kernels running on the GPU.

Implementation 1 [14] is the algorithm employed by the nVidia CUDA Compiler (nvcc) driver. The basis of the algorithm requires us to calculate:

- the magic number
- required number of right shifts
  - is it a special case requiring an additional add operation on the CPU
- send the data to the GPU

However this algorithm requires that both numerator and denominator be represented as same width integers (in our case 64-bit). This presents a problem since GPUs map 64-bit instructions to multiple native 24-bit or 32-bit instructions depending on their compute capabilities. Also the algorithm has two distinct ways of operation depending on whether the divisor and its magic number require an additional add operation which complicates the code and require either an if statement in the kernel code or two separate kernels.

Algorithm 2 First implementation of modular multiplication

1: $mag, shift, add$ ▷ calculate magic number, shift amount and if an addition is required before sending the prime to the GPU
2: **procedure** NVCC_MUL_MOD($a, b, p, mag, shift, add$)
3: $lo = a * b$ ▷ 32 LSB
4: $hi =$ umulhi(lo, mag) ▷ 32 MSB
5: **if** add **then**
6: $sub = lo - hi$
7: $shr = sub \gg 1$
8: $addition = shr + hi$
9: $shr = addition \gg shift$
10: $mul = shr * p$
11: **else**
12: $shr = hi \gg shift$
13: $mul = shr * p$
14: **end if**
15: **return** $(lo - mul)$ ▷ remainder
16: **end procedure**

Implementation 2 [15] is a much more promising algorithm which utilizes the double precision floating point operations (available in devices with compute capability 1.3 and up) as a way to speed up the modulus calculation. One should note that this algorithm allows only up to full-precision 31-bit modular arithmetic. The algorithm also utilizes the magic number 3^{51} [16] as a replacement for a truncate function. When compiled, it uses 14 registers for both GTX 285 and GTX 780.

As an example, we give a simplified implementation of modular multiplication for Kepler (Fermi) architecture from [15].

Algorithm 3 Second implementation of modular multiplication

1: $inv = (\text{double})(1 \ll 30)/p$ ▷ calculate inv before sending the prime to the GPU
2: **procedure** MUL_MOD(a, b, p, inv)
3: $hi =$ umulhi$(a * 2, b * 2)$ ▷ 32 MSB
4: $rf =$ truncate$(hi * inv)$ ▷ $rf = \lfloor r/p \rfloor$
5: $r = a * b - rf * p$ ▷ partial residue
6: **return** $(r < 0\ ?\ r + p : r)$ ▷ adjust by p if negative
7: **end procedure**

Implementation 3 [18] is an improvement of [17]. This algorithm allows us to expand the modulus operation from 2^{31} to 2^{32} and as such it is a supplement to the previous algorithm if one wishes to test up to 2^{32}. This algorithm also seeks to minimize the use of 64-bit arithmetic in favor of 32-bit arithmetic. One of the differences compared to the original algorithm is the fact that we have removed the calculation of the quotient since it is not required in our case. Using **unsigned** variables gives us automatic $(\mathrm{mod}\ 2^{32})$ required by some parts of the algorithm. As some parts of the algorithm need 64-bit variables, we have used lines 4 and 3 to note the required type and size of the variables. When compiled it uses 17 registers for both GTX 285 and GTX 780. We would have tested the mentioned range using this algorithm, however we were time limited in regards to the GPU hardware access.

IV. Results

Using nVidia CUDA GPU Occupancy Calculator found in [10] we can see that register usage for both Implementations 2 and 3 gives us 100% occupancy for the following block sizes:

- GTX 285 - 192, 256, 384, 512
- GTX 780 - 128, 256, 512, 1024.

After testing we have found the differences between the block sizes to be under 1232 milliseconds with the larger block sizes being faster. Given the previous results we have decided on the following: results r_p are received in blocks; the size of a block is obtained by multiplying the number of Streaming Multiprocessors that are used by the number of threads per block, which is chosen to be equal to maximum number of threads per block. The size of a block is chosen to be $SB_1 = 30 \cdot 512 = 15360$ and $SB_2 = 12 \cdot 1024 = 12288$ for GTX 285 and GTX 780, respectively.

Let p_i denote the i-th prime, $p_1 = 2$, and let $\pi(x) = \max\{i \mid p_i \leq x\}$. Let $a = \pi(10^9) + 1 = 50847535$, $b = \pi(2^{31}) + 1 = 105097565$ and $c = \pi(2^{32}) = 203280221$. Then $p_a = 10^9 + 7$, $p_b = 2^{31} + 11$ and $p_c = 2^{32} - 5$. Let $P(u, S) = \{p_u, p_{u+1}, \ldots, p_{u+S-1}\}$ denote the set of S primes starting with p_u. Let $SP_{j,k}$ denote sets used for performance

Algorithm 4 Third implementation of modular multiplication

```
1: rec                                   ▷ calculate reciprocal before sending the prime to the GPU
2: procedure MUL_MOD(a, b, p, rec)
3:     unsigned 32-bit q_1, q_0, r, u_1, u_0          ▷ 32-bit vars
4:     unsigned 64-bit ab, q_1 q_0                    ▷ 64-bit vars
5:     ab = a * b                                     ▷ 64-bit product
6:     u_1 = ab ≫ 32                                  ▷ 32 MSB
7:     u_0 = ab                                       ▷ 32 LSB
8:     q_1 q_0 = rec * u_1                            ▷ 64-bit product
9:     q_1 q_0 = q_1 q_0 + ab
10:    q_1 = q_1 q_0 ≫ 32                             ▷ 32 MSB
11:    q_0 = q_1 q_0                                  ▷ 32 LSB
12:    q_1 = q_1 + 1          ▷ unsigned gives us (mod 2^32)
13:    r = u_0 − (q_1 * p)    ▷ unsigned gives us (mod 2^32)
14:    if r > q_0 then                                ▷ unpredictable condition
15:        r = r + p          ▷ unsigned gives us (mod 2^32)
16:    end if
17:    if r ≥ p then                                  ▷ unlikly condition
18:        r = r − p
19:    end if
20:    return r                                       ▷ remainder
21: end procedure
```

TABLE II
AVERAGE RUNTIME IN SECONDS OVER 20 RUNS.

	GTX 285				GTX 780			
	$SP_{1,1}$	$SP_{1,2}$	$SP_{1,3}$	$SP_{1,4}$	$SP_{2,1}$	$SP_{2,2}$	$SP_{2,3}$	$SP_{2,4}$
I1	24829	53303	53328	106640	717	1531	1566	3203
I2	1891	4126	-	-	428	911	-	-
I3	-	-	5584	12970	-	-	964	2081

TABLE III
AVERAGE SPEEDUP.

	GTX 285				GTX 780			
	$SP_{1,1}$	$SP_{1,2}$	$SP_{1,3}$	$SP_{1,4}$	$SP_{2,1}$	$SP_{2,2}$	$SP_{2,3}$	$SP_{2,4}$
I1	1x	1x	1x	1x	1x	1x	1x	1x
I2	13.13x	12.92x	-	-	1.68x	1.72x	-	-
I3	-	-	9.54x	8.22x	-	-	1.59x	1.54x

comparison of various implementations: $SP_{j,1} = P(a, SB_j)$, $SP_{j,2} = P(b - SB_j + 1, SB_j)$, $SP_{j,3} = P(b, SB_j)$, $SP_{j,4} = P(c - SB_j + 1, SB_j)$, where $j = 1, 2$ for GTX 285 and GTX 780 respectively.

The results are given in Table II (average runtime in seconds over 20 runs over blocks), Table III (relative speedups of Implementations 2 and 3, compared to Implementation 1), and Table IV (average time to calculate r_p for a single prime p, calculated as $time/SB$ in ms). The rows labeled I1, I2, I3 correspond to the three previously described implementations.

As can be seen from Table II because GTX 285 lacks proper 32-bit arithmetic Implementation 1 is not ideal, while GTX 780 gives respectable performance for such implementation. Table III clearly shows that while using different implementations gives massive performance improvements for GTX 285, improvements are more modest for GTX 780. Table IV merely states that due to the way GPUs operate, a single threaded program would have to process $\mathbf{LeftfactorialMod}(p)$ for one prime in a given time frame in order to be competitive.

TABLE IV
AVERAGE TIME TO CALCULATE r_p FOR A SINGLE PRIME p, CALCULATED AS $time/SB$ IN MS.

	GTX 285				GTX 780			
	$SP_{1,1}$	$SP_{1,2}$	$SP_{1,3}$	$SP_{1,4}$	$SP_{2,1}$	$SP_{2,2}$	$SP_{2,3}$	$SP_{2,4}$
I1	1616	3470	3471	6942	58	124	127	260
I2	123	268	-	-	34	74	-	-
I3	-	-	363	844	-	-	78	169

TABLE V
THE VALUES OF $!p$ FROM 10^9 UP TO 2^{31} CLOSE TO 0 OR p.

p	r_p	p	$p - r_p$
1023141859	96	1278568703	6
1167637147	67	1330433659	75
1250341679	15	1867557269	42
1283842181	80		

The results are given in Table V, where the values of r_p and $p - r_p$ satisfying $r_p \leq 100$ or $r_p \geq p - 100$, respectively, are listed, accompanied by the corresponding primes p.

These values were checked using Wolfram Mathematica 9.0.1.0 and are correct according to the implementation of the $\mathbf{LeftfactorialMod}(p)$ given in algorithm 1.

It should be noted that the calculations were done on commercial hardware which lacks Error-correcting code (ECC) memory and zero error tolerance stress testing present in enterprise class hardware. ECC memory is capable of detecting and correcting data corruptions due to internal (magnetic or electrical interference) or external factors (background radiation). ECC memory can correct single bit errors using Hamming codes or triple modular redundancy.

Given the lack of ECC we have decided to validate our results by statistical sample using the formula given by [19] in order to reach the following values: Population size = 54250031, Confidence = 99.9%, Margin of error = 0.5%, Sample size = 108060.

After completion we have randomly chosen 108060 elements and calculated their values on two different central processing units (CPUs). In our validation we have found no errors while comparing the random subset and this works results.

V. COUNTEREXAMPLES TO CONJECTURES ON GENERALIZED LEFT FACTORIALS

It is easily proved that the numbers $K_m(n)$ (4) satisfy the recurrent relation

$$K_{m-1}(n) + K_m(n-1) = K_m(n)$$

implying

$$\sum_{i=0}^{n-1} K_{m-1}(i) = K_m(n)$$

and

$$K_m(n) = \sum_{i=0}^{n-1} \binom{n-i+m-1}{m} i!$$

Consider first the conjecture (5). Suppose $p|\gcd(K_0(n), K_1(n))$ for some prime $p > 5$ and for some n. There are two cases:

- If $n \geq p$ then $K_0(n) \equiv \sum_{i=0}^{p-1} i! \equiv r_p \pmod p$, and $K_1(n) \equiv \sum_{i=0}^{p-1}(n-i)i! = \sum_{i=0}^{p-1}(n-(i+1)+1)i! = n\sum_{i=0}^{p-1} i! - p! + 1 \equiv nr_p + 1 \pmod p$. Consequently, if $K_0(n) = 0 \pmod p$ then $K_1(n) \equiv 1 \neq 0 \pmod p$, and $K_0(n) \equiv K_1(n) \equiv 0 \pmod p$ has no solutions.
- If $n < p$ then $K_0(n) = !n \pmod p$, $K_1(n) = n\cdot!n - n! + 1 \pmod p$ and the system $K_0(n) \equiv K_1(n) \equiv 0 \pmod p$ is equivalent to the system $!n = 0 \pmod p$, $n! = 1 \pmod p$. Searching for pairs (n, p) satisfying this system is similar to Algorithm 1: for fixed prime p one has to check for all $n < p$ if $!n = 0 \pmod p$, $n! = 1 \pmod p$ is true. The first such pair is $(n, p) = (372085, 425701)$, i.e. $372085! = 1 \pmod{425701}, !372085 = 0 \pmod{425701}$.

Hence, the smallest counterexample is $425701|\gcd(K_0(372085), K_1(372085)$.

Consider now the conjecture (6). Suppose $p|\gcd(K_1(n), K_2(n))$ for some prime $p > 5$ and for some $n \geq p$. Then

$$K_1(n) \equiv \sum_{i=0}^{p-1}(n-i)i! \pmod p$$

is a linear polynomial in n and

$$K_2(n) \equiv \sum_{i=0}^{p-1} \binom{n-i+1}{2} i! \pmod p$$

is a quadratic polynomial in n. Hence, it is straightforward to check if $K_1(n) \equiv K_2(n) \equiv 0 \pmod p$ has some solution n. The first p leading to such an counterexample is $p = 859$; then

$$K_1(n) = 1 + 261n \pmod{859},$$

$$K_2(n) = 300 + 561n + 560n^2 \pmod{859}$$

and the system $K_1(n) \equiv K_2(n) \equiv 0 \pmod{859}$ has the solutions $n = 260 + 859k, \quad k \in \mathbb{N}$. The condition $n \not\equiv 9 \pmod{13}$ is equivalent to $k \not\equiv 9 \pmod{13}$. Hence there are many counterexamples to (6), and the smallest is obtained for $n = 260 + 859 = 1119$: $859|\gcd(K_1(1119), K_2(1119))$.

VI. Conclusion

In this work we used several implementations of modular multiplication on GPU in order to check Kurepa's left factorial conjecture for primes up to 2^{31}.

However we have also shown that while it is possible to achieve an improvement, the success is dependent on the GPU architectures. While even earlier devices with compute capability 1.3 are usable for this sort of calculation, the lack of actual 32-bit arithmetic is a major problem.

Our long term goal is to resolve the range of primes up to 2^{32}. Should we use only one GTX 780 we can see that we have to solve 7991 blocks rising to an approximate runtime of 140 days. However we should also strive to potentially change the approach and algorithm used to a more suitable one (utilizing shared memory, minimizing thread divergence and seeking to replace multiplication dependent algorithms with a combination of multiplication and addition algorithms.

While researching Kurepa's left factorial we have found several counterexamples with regards to conjectures on generalized Kurepa's left factorials.

Acknowledgment

We would like to thank Prof. Miodrag Živković who helped devise the premise of this work and for his helpful comments and suggestions and Prof. Gradimir V. Milovanović for informing us about generalization of Kurepa's left factorial and his conjectures.

We would also like to thank various other persons who have been gracious enough to grant us the use of the hardware we utilized in our work.

References

[1] Kurepa Đ., "On the left factorial function," *Math. Balkanica*, 1971, pp. 147-153

[2] Guy R., "Unsolved problems in number theory, 3rd edition," *Springer-Verlag*, 2004, pp. 175

[3] Ivić A., Mijajlović Ž., "On Kurepa's problems in number theory," *Publ. Inst. Math. (Beograd)*, 1995, pp. 19–28

[4] Mijajlović Ž., "On some formulas involving !n and the verification of the !n hypothesis by use of computers," *Publ. Inst. Math. (Beograd)*, 47(61), 1990

[5] Milovanović G. V., "A sequence of Kurepa's functions," *Scientific Review*, 19-20, 1996, pp. 137-146

[6] Živković M., "Massive computation as a problem solving tool," Proceedings of the 10th Congress of Yugoslav Mathematicians (Belgrade, 2001), *Univ. Belgrade Fac. Math.*, Belgrade, 2001

[7] Gallot Y., "Is the number of primes $\frac{1}{2}\sum_{i=0}^{n-1} i!$ finite ?," http://yves.gallot.pagesperso-orange.fr/papers/lfact.html, 20. July. 2000

[8] Jobling P., "A couple of searches," http://groups.yahoo.com/neo/groups/primeform/conversations/topics/5095, 2. December. 2004

[9] Tatarević M., "Searching for a counterexample to the Kurepa's left factorial hypothesis (p < 10^9)," http://mtatar.wordpress.com/2011/07/30/kurepa/, 30. July. 2011

[10] -, "CUDA Toolkit | NVIDIA Developer Zone," *NVIDIA Corporation*, https://developer.nvidia.com/cuda-toolkit

[11] -, "CUDA C Programming Guide 5.5," *NVIDIA Corporation*, 2013

[12] -, "GeForce GTX 285 | GeForce," *NVIDIA Corporation*, http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-285

[13] -, "GeForce GTX 780 Graphics Card with Kepler Technology | GeForce | GeForce," *NVIDIA Corporation*, http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-780

[14] Henry S. Warren Jr., "Hacker's Delight, Second Edition," *Addison-Wesley Professional*, 2012, pp. 227-234

[15] Emeliyanenko P., "Computing resultants on Graphics Processing Units: Towards GPU-accelerated computer algebra," Journal of Parallel and Distributed Computing, 2013, pp. 1494-1505

[16] Hecker C., "Let's get to the (floating) point," Game Developer Magazine, 1996, pp. 19-24

[17] Granlund T., P. L. Montgomery, "Division by invariant integers using multiplication," Proceedings of the ACM SIGPLAN 1994 conference on Programming language design and implementation, 1994, pp. 61-72

[18] Möller N., Granlund T., "Improved Division by Invariant Integers," IEEE Transactions on Computers, 2011, pp. 165-175

[19] Krejcie R. V., Morgan D. W. "Determining Sample Size for Research Activities," Educational and Psychological Measurement 30, 1970, pp. 607-610

Finding Collision for MD4 Hash Algorithm Using Hybrid Algorithm

Carić Marko

Abstract - ***The modification of a message that meets sufficient conditions for collision is found in the last step of differential attack proposed by Wang et all. (2005) on MD4 hash algorithm. In this article we show how this attack phase, finding a collision starting from the list of sufficient conditions for the collision, can be implemented using a combination of two algorithms – evolutionary algorithm and hill climbing. Hybridization of evolutionary algorithm and hill climbing is a well-known technique for improving solutions, but it hasn't been applied to this domain (at least according to informations that the author has collected).***

Key words - ***Evolutionary algorithm, hill climbing, hash algorithm, MD4, collision, differential attack***

1. INTRODUCTION

A *cryptographic hash algorithm* for arbitrary message *M* computes fixed-size bit string *hash value h(M)* such that any change of *M* significantly must change *h(M).* For example, the hash algorithm MD4 [5] for an arbitrary message computes its hash value of 128 bits in length. The request that a good hash algorithm should meet is that for a given message *M* it is practically impossible to find another message *M'* such that *h(M) = h(M').* Often, stricter requirement is set it should be virtually impossible to find a pair of messages *(M, M')* (*collision*) such that *h(M) = h(M').* In the case of *differential attack* on a hash algorithm, we follow the evolution of two similar messages that pass through the algorithm. Differential attack on a hash algorithm can be divided into four phases:

- Selection of message difference,
- Finding the differential path,
- Finding sufficient conditions for collision,
- Finding messages that satisfy sufficient conditions.

The first two stages can be automated to some extent [8,9]. One way of realizing the last phase is to gradually modify randomly selected message. The modification of the message in different rounds of the MD4 algorithm is different in terms of difficulty. Modification to satisfy the conditions of the first round is the easiest; it requires changing one message word (*basic modification).* Modification to satisfy the conditions of the second round is more complicated; as it must not disarrange the already fulfilled conditions of the first round, it is necessary to change more than one word of the message (*advanced modification*). Typically, a brute force is used to find modification satisfying the conditions of the third round. That is why the tendency is to include most conditions for modification of the message in the first round [3]. One way to realize message modification is achieved using SAT solvers [6]. Based on author's knowledge, there is no other effective way for solving this step.

This paper proposes a possibility of the realization of the last phase of the attack independently either by using evolutionary algorithms (EA) or by hill climbing, or by their combination. Alternating EA and hill climbing can improve the efficiency of EA while overcoming the lack of robustness of hill climbing.

Manuscript received on October 23th, 2013.

Marko Carić is with The School of Electrical and Computer Engineering of Applied Studies Belgrade (e-mail: caric.marko@gmail.com).

2. DESCRIPTION OF MD4

Let $B=\{0,1\}$. MD4 algorithm for arbitrary length message M calculates the hash value $h(M) \in B^{128}$. An iterative compression function $f: B^{128} \times B^{512} \rightarrow B^{128}$ is defined using functions $F,G,H: B^{32} \times B^{32} \times B^{32} \rightarrow B^{32}$:

$$F(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z)$$
$$G(X,Y,Z) = (X \wedge Y) \vee (X \wedge Z) \vee (Y \wedge Z)$$
$$H(X,Y,Z) = X \oplus Y \oplus Z$$

where $\wedge, \vee, \neg, \oplus$ are bit operations AND, OR, NEG, XOR applied to the corresponding 32 bits of the arguments. Let $X+Y$, $X-Y$ denote the sum and difference X and Y modulo 2^{32}, and let $X << s$ be the result of the cyclical movement of the contents of X for s bits to the left.

Let $t_i \in B^{32}$, $i=-3,-2,\ldots,48$, $T=(t_{-3}, t_{-2}, t_{-1}, t_0) \in B^{128}$ i $M=(M_0, M_1, \ldots, M_{15}) \in B^{512}$. Then $f(T,M)= (t_{-3}+t_{45}, t_{-2}+t_{46}, t_{-1}+t_{47}, t_0+t_{48})$, where

$$t_i=(t_{i-4}+f_i(t_{i-3},t_{i-2},t_{i-1})+M_{p(i)}+c_i)<<s_i,\ i=1,2,\ldots,48, \quad (1)$$

and values f_i, c_i, p_i, s_i for $i=1,2,\ldots,48$ are shown in the following table:

i	f_i	c_i	p_i	s_i
1,...,16	F	0000 0000	0,1,2,3,4,5,6,7,8 ,9,10,11, 12,13,14,15	3,7,11,19,3,7, 11,19,3,7,11, 19,3,7,11,19
17,...,32	G	5a82 7999	0,4,8,12,1,5, 9,13,2,6,10, 14,3,7,11,15	3,5,9,13,3,5, 9,13,3,5,9,13, 3,5,9,13
33,...,48	H	6ed9 eba1	0,8,4,12,2, 10,6,14,1,9,5, 13,3,11,7,15	3,9,11,15,3,9, 11,15,3,9,11, 15,3,9,11,15

Calculation of t_i, $i=16(j-1)+k$, $k=1,2,\ldots,16$ (1) makes the jth *round* of MD4, $j=1,2,3$.
Before calculating hash value, the message M is first padded to the multiple of 512 by one, $447-|M| \bmod 512$ zeroes (where $a \bmod b \in \{0,1,\ldots,b-1\}$ denotes the remainder of division of b by a) and 64 bits to represent a binary number $|M|$. Let expanded message be $M_0, M_1, \ldots, M_{n-1}$, $M_i \in B^{512}$, $i=1,2,\ldots,n-1$. The hash value of the message M is $h(M) = H_n$, where
H_0=(67452301,10325476,efcdab89,98badcfe), (2) (hex), and $H_i=f(H_{i-1},M_{i-1})$, $i=1,2,\ldots,n$.

3. DIFFERENTIAL PATH AND SET OF SUFFICIENT CONDITIONS

To find collisions for MD4 algorithm it is enough to find collisions for the compression function f, i.e. an arbitrary pair (M, M'), $M=(M_0, M_1, \ldots, M_{15})$, $M'=(M'_0, M'_1, \ldots, M'_{15})$, $(M, M') \in B^{512}$ such that $f(H_0,M) = f(H_0,M')$, where H_0 is a constant (2). This equation is equivalent to $(t_{45},t_{46},t_{47},t_{48})=(t'_{45},t'_{46},t'_{47},t'_{48})$, where, according to (1)
$t_i=(t_{i-4}+f_i(t_{i-3},t_{i-2},t_{i-1})+M_{p(i)}+c_i)<<s_i$, $t'_i=(t'_{i-4}+f_i(t'_{i-3},t'_{i-2},t'_{i-1})+M'_{p(i)}+c_i)<<s_i$, $i=1,2,\ldots,48$. (3)

The *message difference* corresponding to the collision (M, M') is a series of differences (modulo 2^{32}) $\Delta M=M'-M=(\Delta m_0, \Delta m_1,\ldots, \Delta m_{15}) = (M_0' - M_0, M_1' - M_1, \ldots, M_{15}' - M_{15})$.
Let $t_i[j]$i $t_i[-j]$ be the values obtained from t_i by changing the jth bit; thereby, $t_i[j]$ means that the value of the jth bit is changed from zero to one, while $t_i[-j]$ means that the value of the jth bit is changed from one to zero. More generally, let $t_i[\pm j_1,\pm j_2,\ldots,\pm j_l]$ be obtained by changing the values of bits in a indicated indexes. The *differential path* is a series of expressions of the form
$t'_i = t_i[\pm j_1,\pm j_2,\ldots,\pm j_l]$, $I=I(i)$, $i=1,\ldots,48$, (4)
where $I=0$ for $i=45,46,47,48$. In order for a pair (M, M') to be a collision, a set of *sufficient conditions* for a pair (M, M') is determined by the differential (4) and conditions (3). This can be obtained starting from the properties of functions F, G and H. A message M that satisfies a set of sufficient conditions is called a *weak message*. If M is a weak message and $M'=M+\Delta M$, then the pair (M, M') is a collision. In this paper we use two message differences: $\Delta m_4=2^2$, $\Delta m_i=0, i\neq 4$ [1,11] and $\Delta m_1=2^{31}$, $\Delta m_2= 2^{31}-2^{28}$, $\Delta m_{16}=-2^{16}$, $\Delta m_i=0, i\neq 1,2,16$ [4,9]. The first message difference corresponds to a set of 62 conditions [11], that is obtained starting from the differential path that consists of two branches:

$$t_5[6] \to t_9[-9,10] \to t_{13}[12] \to t_{17}[15] \to$$
$$t_{18}[20] \to t_{22}[25] \to t_{26}[30] \to t_{30}[3]$$

and

$$t_5[6] \to t_9[-9] \to t_{11}[-20] \to t_{15}[-31] \to$$
$$t_{19}[-8] \to t_{23}[17,-18] \to t_{27}[-26] \to t_{31}[-3]$$

This differential path and the corresponding 62 sufficient conditions for message M are shown in Table 3.
The difference $t_{19}[-8]$ cancels the difference $t_{18}[8]$ caused by the influence of Δm_4 in the second round. The difference $t_{23}[-18]$ cancels difference that occurs as a continuation of the changes in the first round. Similarly, if sufficient conditions are satisfied in steps:

$t_{32}=(t_{28}+G(t_{31},t_{30},t_{29})+M_{15}+5a827999)<<13$,
$t_{33}=(t_{29}+H(t_{32},t_{31},t_{30})+M_0+6ed9eba1)<<3$,
$t_{34}=(t_{30}+H(t_{33},t_{32},t_{31})+M_8+6ed9eba1)<<9$,
$t_{35}=(t_{31}+H(t_{34},t_{33},t_{32})+M_4+6ed9eba1)<<11$,

then the differences cancel each other out in terms of t_{32}, t_{33} i t_{34}, while the difference $t_{31}[-3]$ cancels the last changed word M_4 and the pair $(M,M+\Delta M)$ is collision. Second message difference corresponds to the a set of 123 (in the original paper [4] the conditions listed 122, and the work [7] shows that the two conditions are omitted, it turns out that one of the conditions expressed by the other two, so the number of sufficient conditions is 123), e.g. 146 sufficient conditions [9].

4. ALGORITHM FOR FINDING COLLISIONS

To search for collisions by evolutionary algorithm, or by hill climbing, it is necessary to choose a fitness function, whose optimization leads to a message M for which $(M,M+\Delta M)$ is collision. For this purpose, the fitness function $c(M)$ is defined as maximal k such that the first k sufficient conditions are satisfied for the sequence in (1). The rest of this section presents an evolutionary algorithm, a hill climbing algorithm and a hybrid algorithm, which is actually their composition.

4.1 Evolutionary algorithm

A population which is input to the evolutionary algorithm is made of N randomly selected messages $p[1], p[2],\ldots,p[N]$. Individuals are considered to be the messages that need to be changed to meet the sufficient conditions for the collision. Individual fitness function is defined in terms of set of sufficient conditions that an individual meets. Individual crossover is defined

as a replacement of the parts of messages, and the individual mutation means changing a few of their bits. Since the main purpose of the evolutionary algorithm is successive message modification in a way that the conditions for a collision are successively fulfilled, this paper uses its modified version where the process of selection involves comparing the existing individuals with new ones and the survival of the fittest. This approach is the result of experiments in which a standard genetic algorithm that allows the possibility that two weak individuals can produce a good one, gave worse results. This can be explained by a choice of fitness function for which all conditions must be successively met, making it unlikely that the two low fitness individuals may produce a good one. To the lack of possibility for inapt individuals to survive, leads to uniformity of the population. As the individuals become very similar, enhanced mutation is applied in each iteration. In this way, the evolutionary algorithm actually simulates message modification, which tends to incline towards the message that meets all relevant conditions. This process can be optionally continued using hill climbing algorithm. A general evolutionary algorithm can be described by the following code:

Algorithm EA(*p,N,M***)**
Input: *p* - initial population, a series of *N* messages
M - number of generations in EA
Output: *p* - the final population after generation *M*
Begin
  for *i*←1 to *M* **do**
    for *j*←1 to *N*/4 **do**
      if(*p*[2*j*-1]=*p*[2*j*]) **then** mutation (*p*[2*j*])
      crossover *p*[2*j*-1], *p*[2*j*]
        creates *p*[*N*+2*j*-1], *p*[*N*+2*j*]
      mutation (*p*[*N*+2*j*-1], *p*[*N*+2*j*])
    sort *p* by values *c*(*p*[*j*])) and keep the
        first *N* messages
    for *j*←2 to *N* **do**
      while (*p*[*j*] is equal *p*[*k*] for some
        k=1,2,..., *j*-1) **do** mutation (*p*[*j*])
  return *p*
End

EA uses elimination selection without duplicates. In each iteration, we crossover messages *p[2i-1]* and *p[2i]* with random crossing point resulting in their children, the messages *p[N+2i-1]* and *p[N+2i], i*=1,2,...,*N*/4.
The crossover is done in a standard way:

$$(\alpha,\beta),(\gamma,\delta)\rightarrow(\alpha,\delta),(\gamma,\beta),$$

$$0\le|\alpha|=|\gamma|\le 512$$

After each iteration, all individuals are sorted according to fitness function value. The individuals with lower fitness are rejected in a way the population size remains unchanged. The fittest individuals do not fall out of the population. Mutation of individuals is carried out in three cases: to avoid the crossover of the same message, in order to further distinguish parents from the children and after changing population to throw out possible duplicate messages. Mutation is performed in the following way: choose a random number 0≤*n*≤127, and xor message bits in positions 4*n*,4*n*+1,4*n*+2,4*n*+3 with random 4-bit block. In that way at least one of the bits in the specified positions will be inverted with probability 15/16.

4.2 Correction algorithm

Let $d(x,y)$ denote the Hamming distance between vectors *x* and *y*. For arbitrary $y\in B^{512}$ let $S_t(y)=\{x\mid d(x,y)\le t\}$ denote the Hamming ball of radius *t* with center *y*. If the fittest messages from EA does not satisfy all sufficient conditions, we continue with correction algorithm (CA) - modification of the "Steepest-ascent hill climbing" algorithm [10]. The fittest messages obtained from the evolutionary algorithm are inserted into the priority queue *Q*. Correction algorithm in each iteration removes from *Q* the message *y* of the highest priority, computes *c(z)* for all $z\in S_t(y)$ and inserts into *Q* all *z* for which *c(z)* ≥ *c(y)*. The algorithm is described by the following code:

Algorithm CA(*Q,MAX,OP***)**
Input: *Q* - array of messages
MAX - maximum possible message fitness
OP - limit for the number of tested messages achieving maximum fitness
Output: *collisions* – array of weak messages
Begin
  collisions ← Ø
  counter ← 0
  y ← pullHighestPriorityElement(*Q)*
  achievedFitness ← *y.fitness*
  while(*y.fitness*<MAX **and** *counter* < *OP*)
    for all $p\in S_3(y)$ **do**
      if *p* is new **and** *p.fitness* ≥ *achievedFitness*
      then
        achievedFitness ← *p.fitness*
        Q ← *p*
      if *achievedFitness* = *y.fitness* **then**
        increment *counter*
      else
        counter ← 0
      y←pullHighestPriorityElement(*Q)*
  collisions ← { *p* | *p* ∈ *Q*, *p.fitness* = *MAX*}
  return *collisions*
End

If the number of subsequent messages *y* with the same fitness less than *MAX* exceeds *OP*, the execution stops.

4.3 Hybrid algorithm

Hybrid algorithm is the composition of EA and

CA.

Algoritam HA(*N,M,MAX,OP,V***)**
Input: *N* - initial size of the population in EA
M - number of generation in EA
MAX - maximum possible message fitness
OP - limit for the number of tested messages which achieve maximum fitness
V - initial size of the queue for CA
Output: *collisions* – array of the weak messages
Begin
collisions ← Ø
Q ← Ø
p ← set of *N* random messages
newPopulation ← EA(*p*,*N*,*M*)
collisions ← all weak messages from *newPopulation*
if(*collisions* ≠ Ø) **then**
return *collisions*
else
Q ← first *V* messages from *newPopulation*
return CA(*Q,MAX,OP*)
End

5. RESULTS

The population size *N* = 800 and the number of generations *M* = 1000 were chosen for the evolutionary algorithm. For the *OP* parameter used in correction algorithm, the value *OP* = 7 was selected. The results of 10 tests of the first, second and the third set of sufficient conditions (with successively 62, 123 and 144 conditions) are shown in Table 1.

Table 1. Test results

Test number	62 conditions		123 conditions		144 conditions	
	EA	HA	EA	HA	EA	HA
1	56	62	*123**		*144**	
2	54	62	*123**		138	144
3	52	62	123		130	144
4	55	62	116	123	144	
5	54	62	104	123	144	
6	51	62	115	115	143	*144**
7	54	62	105	105	140	144
8	52	62	116	116	*144**	
9	51	56	99	99	144	
10	59	59	115	115	133	*144**

In the table, the number written in italic font indicates the case when all the conditions are met, but the collision was not found. Since EA cannot always find a collision when the second set of 123 conditions is met, it can be concluded that this set of sufficient conditions derived from the differential path [4] is not complete. In the case of the third set of 146 conditions, the paper [9] gives an incomplete list of 144 of them, and their fulfillment by EA does not always produce the expected collision, since the omitted conditions are randomly met. Generally, if *m* conditions are omitted from a set of sufficient conditions, then to obtain a collision, EA needs to create 2^m weak messages on average. Specifically, for a population of 800 messages, after 1000 iterations, in the best case, the whole newly obtained population (of the last generation) satisfies all the conditions; then for $m \leq 9$ EA continues to produce collisions. The results show that EA successfully produced collisions with an incomplete set of sufficient conditions, which means that EA can also be used as a criterion to check the completeness of a set of sufficient conditions. The result of the experiment is quite satisfactory.

Tests were carried out on a computer with a processor AMD Phenom II X4 945 3.01 GHz and 4.00 GB of working memory. Evolutionary algorithm runs about ten minutes for all sets of conditions, while CA examines one message about 20 minutes. Although CA is designed to serve as a possible continuation of the EA, it is also independent of EA and can independently produce collisions. However, on average, it takes more time and produces fewer collisions than EA. From an initial population of randomly generated 800 messages, the fittest of them on average meets the first 9 conditions (probability of satisfying these collisions is $1/2^9$), after which a message that satisfies all sufficient conditions is created by successive corrections

As an illustration of the results obtained, the examples of collisions detected for each of the three used sets of sufficient conditions are given. Messages *M* and *M'* are different in the third bit of the fourth word in the case of 62 conditions, or in bits of the first, second and twelfth word in the case of 123 and 146 conditions (Table 2). The above collisions were obtained in experiments numbered by 6,3,4 in Table 1.

Table 2. Examples of collisions for the first, second or third set of sufficient conditions

M	53fdda09	dbd460d2
	6b0d1c7e	d41233e2
	0e973a6**3**	ee35f949
	3d28ca69	6d101738
	1f760241	173b3175
	07531617	4c867a98
	797525e2	dd0b1b98
	cec0df99	4f45d906
M'	53fdda09	dbd460d2
	6b0d1c7e	d41233e2
	0e973a6**7**	ee35f949
	3d28ca69	6d101738
	1f760241	173b3175
	07531617	4c867a98
	797525e2	dd0b1b98
	cec0df99	4f45d906
$f(H_0,M) = f(H_0,M')$	cd9369f2	5c77a88c
	349b7fd3	b820249c

M	e85075c4	**6**e37e13b
	729fb981	ffefea77
	8ae24c2b	cec4f21e
	62a4c566	b681a50a
	89e04800	e9bfdf35
	5d38b88a	9a847d14
	9af**d**08ad	bfdc2bf2
	e6b416e2	d7edc8b5
M'	e85075c4	**e**e37e13b
	e29fb981	ffefea77
	8ae24c2b	cec4f21e

	62a4c566 b681a50a 89e04800 e9bfdf35 5d38b88a 9a847d14 9af**c**08ad bfdc2bf2 e6b416e2 d7edc8b5
$f(H_0,M) = f(H_0,M')$	85a9276a 331f0b93 7661e284 92531baa

M	7d4182c2 **3**64f032b **f**369358a 32568a9e 1417ea70 357ba164 770ab75f bb95772e f78c5797 b651d1fa 234a288d 3660f5c0 004**0**cb07 a9ded82a 458d2286 29378008
M'	7d4182c2 **b**64f032b **6**369358a 32568a9e 1417ea70 357ba164 770ab75f bb95772e f78c5797 b651d1fa 234a288d 3660f5c0 004**f**cb07 a9ded82a 458d2286 29378008
$f(H_0,M) = f(H_0,M')$	eb8fa12e cf298b09 bcec958d 4cefae4b

Experiments with a fitness function equal to the total number of satisfied conditions did not give good results, except for the first, the smallest set of sufficient conditions.

6. COMPARISON WITH THE OTHER APPROACHES

Message modification problem, which has been solved using SAT solvers [6] and which also gives collision in less than 10 minutes, is a somewhat faster solution as it has been obtained in a computer with a low performance. However, the heuristic approach discussed in this paper produces not one but multiple collisions. In the best case, if the evolutionary algorithm succeeds in producing collisions on its own, each message in the population may produce a collision.

7. CONCLUSION

After obtaining sufficient conditions for MD4 collision, message modification can be very demanding. This work explains a new method for replacing standard message modification phase in MD4 algorithm using evolutionary algorithm and hill climbing. Further research involves the application of this procedure to other hash algorithms belonging to the MD4 family.

REFERENCES

[1] H.B. Yu, G.L. Wang, G.Y. Zhang, X.Y. Wang: *The Second-Preimage Attack on MD4*. CANS 2005, LNCS 3810, pp. 1-12.

[2] X.Y. Wang, H.B. Yu: *How to Break MD5 and Other Hash Functions*. Eurocrypt'05, LNCS 3494, pp.19-35.

[3] Y. Sasaki, L. Wang, K. Ohta, N. Kunihiro: *New Message Difference for MD4*. In A. Biryukov, ed. FSE 2007, Proceedings, volume 4593 of LNCS, pp. 329-348. Springer, 2007.

[4] X.Y. Wang, X.J. Lai, D.G. Feng, H. Chen, X.Y. Yu: *Cryptanalysis for Hash Functions MD4 and RIPEMD*. Eurocrypt'05, May 2005.

[5] R.L. Rivest: *The MD4 Message Digest Algorithm*. Advances in Cryptology, Crypto'90, Springer 1991, pp. 303-311.

[6] I. Mironov, L. Zhang: Applications of SAT Solvers to Cryptanalysis of Hash Functions. SAT 2006, pp.102-115.

[7] Y. Naito, Y. Sasaki, N. Kunihiro, K. Ohta: *Improved Collision Attack on MD4 with Probability Almost 1*. In D. Won and S. Kim, eds. ICISC, volume 3935 LNCS, pp. 129-145. Springer, 2005.

[8] P.A. Fouque, G. Leurent, P. Nguyen: *Automatic Search of Differential Path in MD4*. ECRYPT Hash Worshop – Cryptology ePrint Archive, Report 2007/206 (2007) http://eprint.iacr.org/.

[9] M. Schläffer, E. Oswald: *Searching for Differential Paths in MD4*. Robshaw, FSE 2006, pp. 242–261.

[10] M. Mitchell: *An introduction to genetic algorithms*. MIT Press, Cambridge, Massachusetts (1999).

[11] M. Carić: *Finding collisions in cryptographic hash functions*, Master Thesis, 2010 (in Serbian).

[12] M. Carić, M. Živković: *Finding collisions for MD4 hash algorithm using metaheuristics algorithms,* YUINFO 2013, Kopaonik, pp. 628 – 632 (in Serbian).

Marko Carić received his B.Sc. in mathematics in 2002 and his M. Sc. Degree in computer science in 2010 from the University of Belgrade, Serbia. Since 2002, he is with The School of Electrical and Computer Engineering of Applied Studies.

Table 3. Differential path that corresponds to the message difference $\Delta m_4=2^2$, $\Delta m_i=0, i\neq 4$

i	$M_{p(i)}$	s_i	ΔM_i	$t'_i - t_i$	t'_i	Sufficient conditions
1	M_0	3			t_1	
2	M_1	7			t_2	
3	M_2	11			t_3	
4	M_3	19			t_4	
5	M_4	3	2^2	2^5	$t_5[6]$	$t_{5,6}=0$
6	M_5	7			t_6	$t_{4,6}=t_{3,6}$
7	M_6	11			t_7	$t_{6,6}=0$
8	M_7	19			t_8	$t_{7,6}=1$
9	M_8	3		2^8	$t_9[-9,10]$	$t_{9,9}=1$, $t_{9,10}=0$
10	M_9	7			t_{10}	$t_{8,9}=t_{7,9}$, $t_{8,10}=t_{7,10}$
11	M_{10}	11		-2^{19}	$t_{11}[-20]$	$t_{11,20}=1$, $t_{10,9}=1$, $t_{10,10}=0$
12	M_{11}	19			t_{12}	$t_{11,9}=1$, $t_{11,10}=1$, $t_{10,20}= t_{9,20}$
13	M_{12}	3		2^{11}	$t_{13}[12]$	$t_{13,12}=0$, $t_{12,20}=0$
14	M_{13}	7			t_{14}	$t_{12,12}=t_{11,12}$, $t_{13,20}=1$
15	M_{14}	11		-2^{30}	$t_{15}[-31]$	$t_{15,31}=1$, $t_{14,12}=0$
16	M_{15}	19			t_{16}	$t_{14,31}= t_{13,31}$, $t_{15,12}=1$
17	M_0	3		2^{14}	$t_{17}[15]$	$t_{17,15}=0$, $t_{16,31}= t_{14,31}$
18	M_4	5	2^2	2^7+2^{19}	$t_{18}[8,20]$	$t_{18,8}=0$, $t_{16,15}=t_{15,15}+1$, $t_{18,20}=0$, $t_{17,31}=t_{16,31}$
19	M_8	9		-2^7	$t_{19}[-8]$	$t_{19,8}=1$, $t_{17,8}=t_{16,8}$, $t_{18,15}=t_{16,15}$, $t_{17,20}=t_{16,20}$
20	M_{12}	13			t_{20}	$t_{19,15}= t_{18,15}$, $t_{19,20}= t_{17,20}$
21	M_1	3		2^{17}	$t_{21}[18]$	$t_{21,18}=0$, $t_{20,20}=t_{19,20}$
22	M_5	5		2^{24}	$t_{22}[25]$	$t_{22,25}=0$, $t_{20,18}=t_{19,18}$, $t_{21,8}=t_{20,8}+1$
23	M_9	9		-2^{16}	$t_{23}[17,-18]$	$t_{23,17}=0$, $t_{23,18}=1$, $t_{22,18}=t_{20,18}$, $t_{21,25}=t_{20,25}$
24	M_{13}	13			t_{24}	$t_{22,17}=t_{21,17}$, $t_{23,25}=t_{21,25}$
25	M_2	3			t_{25}	$t_{24,17}=t_{22,17}$, $t_{24,25}=t_{23,25}$, $t_{24,18}=t_{22,18}+1$
26	M_6	5		2^{29}	$t_{26}[30]$	$t_{25,17}=t_{24,17}$, $t_{25,18}=t_{24,18}$, $t_{26,30}=0$
27	M_{10}	9		-2^{25}	$t_{27}[-26]$	$t_{27,26}=1$, $t_{25,30}=t_{24,30}$
28	M_{14}	13			t_{28}	$t_{26,26}=t_{25,26}$, $t_{27,30}=t_{25,30}$
29	M_3	3			t_{29}	$t_{28,26}=t_{26,26}$, $t_{28,30}=t_{27,30}$
30	M_7	5		2^2	$t_{30}[3]$	$t_{30,3}=0$, $t_{29,26}=t_{28,26}$
31	M_{11}	9		-2^2	$t_{31}[-3]$	$t_{31,3}=1$, $t_{29,3}=t_{28,3}$
32	M_{15}	13			t_{32}	
33	M_0	3			t_{33}	
34	M_8	9			t_{34}	$t_{33,3}=t_{32,3}$
35	M_4	11	2^2		t_{35}	

Permutations with Constrained Consecutive k-sums — Some Special Cases

Stefanović, Milan; and Živković, Miodrag

***Abstract*—For a permutation $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ of $1, 2, \ldots, n$ whose elements are written on a circle (and so indices are counted modulo n) we consider sums of all consecutive k elements, the difference $\mathrm{msum}(\pi, k)$ of maximal such sum and the average sum $k(n+1)/2$, and discrepancy $disc(\pi, k)$, the maximum distance of these sums from $k(n+1)/2$. Let $\mathrm{disc}(n, k)$ and $\mathrm{msum}(n, k)$ denote the minimum of $\mathrm{msum}(\pi, k)$ and $\mathrm{disc}(\pi, k)$, over all permutations $\pi \in S_n$. Here we prove that $\mathrm{disc}(6t+3, 3) = 2$, completing previously known results about $\mathrm{disc}(n, 3)$. The value of $\mathrm{msum}(2km \pm 2, 2k) = 1$ is determined by supplying a construction.**

***Index Terms*—permutation, discrepancy, maximal consecutive sum.**

I. Introduction

LET $\pi = (\pi_1, \pi_2, \ldots, \pi_n) \in S_n$ and suppose $\pi_{i+n} = \pi_i$, $i \geq 1$. As in [1] consider subsequent k-sums $s_i = \sum_{j=1}^{k} \pi_{i+j}$, $i \geq 1$, their normalized maximum

$$\mathrm{msum}(\pi, k) = \max_{1 \leq i \leq n} s_i - k(n+1)/2 \tag{1}$$

and their *discrepancy*

$$\mathrm{disc}(\pi, k) = \max_{1 \leq i \leq n} |s_i - k(n+1)/2|; \tag{2}$$

here $k(n+1)/2 = \frac{1}{n}\sum_{i=1}^{n} s_i$ is the average of all subsequent k-sums. Furthermore, let

$$\mathrm{msum}(n, k) = \min_{\pi \in S_n} \mathrm{msum}(\pi, k) \tag{3}$$

and let

$$\mathrm{disc}(n, k) = \min_{\pi \in S_n} \mathrm{disc}(\pi, k). \tag{4}$$

Obviously, $\mathrm{msum}(\pi, k) \leq \mathrm{disc}(\pi, k)$ and $\mathrm{msum}(n, k) \leq \mathrm{disc}(n, k)$.

For example, if $k = 4$, $n = 10$ and $\pi \in S_{10}$ is the permutation

i	1	2	3	4	5	6	7	8	9	10
π_i	9	2	7	4	10	1	8	3	6	5
s_i	22	23	22	23	22	18	22	23	22	23

then $\mathrm{msum}(\pi, 4) = 23 - 22 = 1$.

Anstee et all. [1] derived some upper and lower bounds on $\mathrm{disc}(n, k)$. Given a permutation $\pi \in S_n$, they consider the vector $d = (d_1, d_2, \ldots, d_n)$ whose components are the differences

$$d_i = s_{i+1} - s_i = \pi_{i+k} - \pi_i, \quad i \geq 1.$$

If $1 \leq s \leq t \leq n$ then $\sum_{j=s}^{t} d_j = s_{t+1} - s_s$ is a diffference of some two k-sums. Nearly all discrepancies are determined with $k = 3$ [1]:

- $\mathrm{disc}(n, 3) = 3/2$ for all even $n \geq 8$ (the exception is $\mathrm{disc}(6, 3) = 1/2$),
- $\mathrm{disc}(6t \pm 1, 3) = 2$ (except for $\mathrm{disc}(5, 3) = 1$),
- $\mathrm{disc}(6t+3, 3) \leq 2$ if $t \geq 2$, $\mathrm{disc}(9, 3) = 1$.

Morris [9] and Stefanović [2] determined exact values of $\mathrm{msum}(n, k)$ for small n, k and obtained various bounds. Determination of $\mathrm{msum}(n, 3)$ was a task at programming contest [10]. Rolfe [3], [4] suggested to use the determination of $\mathrm{msum}(n, 3)$ as a replacement of 8-queens problem while teaching students backtracking. An exercise in the discrete math text by Liu [7] is to prove that $\mathrm{msum}(36, 3) \geq 1/2$. In [1], [8] the simpler problem of linear discrepancy is considered, restricting to linear permutations (not reducing indices modulo n). Smallest value of maximum sum of two subsequent terms of permutations is considered in [5], with the additional constraint that sums are distinct. Erdös et all [6] give the asymptotic upper bound on the minimal maximum least common multiple of subsequent terms of a permutation.

Here we prove that if $t \geq 2$, then $\mathrm{disc}(6t+3, 3) > 1$, thus solving the remaining case for $k = 3$. Furthermore, we show that $\mathrm{msum}(2km \pm 2, 2k) = 1$.

II. Simple lower bound

In [1] the fact is used that $\mathrm{disc}(n, 3)$, being less than $3/2$ cannot be $1/2$ if $n \geq 8$. We now derive analogous simple more general lower bound on $\mathrm{msum}(n, k) \leq \mathrm{disc}(n, k)$.

Theorem 1. *If $n > 2k$, then $\mathrm{msum}(n, k) > 1/2$.*

Proof. Let $x = k(n+1)/2$. Note that if $n > k$ then $\mathrm{msum}(n, k) > 0$. Indeed, if $\pi \in S_n$ is such that $s_1 = s_2 = \cdots = s_n = x$, then $0 = d_1 = s_2 - s_1 = \pi_{k+1} - \pi_1$ and π is not a permutation.

If k is even and $n > 2k$, then $\mathrm{msum}(n, k) \geq 1 > 1/2$ is an integer. Now suppose k is odd, $n > 2k$ and let $\pi \in S_n$ be a permutation such that $\mathrm{msum}(\pi, k) = 1/2$, i.e. $s_i \leq x + 1/2$ for all i, $1 \leq i \leq n$. An equality $s_i = s_{i+1}$ would imply $\pi_{k+i} = \pi_i$, hence each two maximal sums s_i (the sums equal to $x + 1/2$) are separated by at least one smaller sum (a sum that is less than or equal to $x - 1/2$). The fact that the maximal and smaller sums are interleaved imply that if the number of

Manuscript received March 14, 2014.

Milan Stefanović, M. Sc. in Mathematics, currently employed in E-Smart Systems company (e-mail: stefke381@gmail.com).

Miodrag Živković, is with the Computer science department, Faculty of Mathematics, Belgrade (e-mail: ezivkovm@matf.bg.ac.rs).

Supported by Serbian Ministry of Science, Grant 174021.

maximal sums is m, then the number $n-m$ of smaller sums is at least m. The opposite inequality $n-m \le m$ follows from the fact that if $n-m > m$, then the average of all s_i would be less than

$$\begin{aligned} & m(x+1/2)/n + (n-m)(x-1/2)/n \\ = \;& x + (2n-m)/2 < x \end{aligned}$$

Therefore $n-m = m$, and the smaller sums are all equal to $x-1/2$. If for example $s_2 - s_1 = -1$, then $\pi_{i+k} - \pi_i = s_{i+1} - s_i = (-1)^i/2$, $i \ge 1$, and $\pi_{2k+1} = \pi_{k+1} + d_{k+1} = \pi_{k+1} + d_{k+1} + d_1 = \pi_1$. This is a contradiction, because $2k+1 \le n$. □

If k is odd, then $\mathrm{disc}(2kt, k)$ is not an integer. Combining $\mathrm{disc}(2kt, k) \le 2$ [1][Theorem 9] and $\mathrm{disc}(2kt, k) > 1/2$, we obtain

Corrolary 2. *If k is odd and $t > 1$, then* $\mathrm{disc}(2kt, k) = \mathrm{msum}(2kt, k) = 3/2$.

III. Determination of $\mathrm{disc}(6t+3, 3)$

Similar, but more elaborate consideration gives the lower bound on $\mathrm{disc}(6t+3, 3)$, equal to the upper bound from [1].

Theorem 3. *Let $n = 6t+3 \ge 15$. Then* $\mathrm{disc}(n, 3) = 2$.

Proof. By [1][Theorem 4] $\mathrm{disc}(n, 3) \le 2$. It remains to prove that if $n = 6t+3 \ge 15$, then there is no permutation $\pi \in S_n$ such that $\mathrm{disc}(\pi, 3) = 1$. Suppose that there is such a permutation π. Let $x = 3(n+1)/2$ denote average 3-sum of π. Then $s_i \in \{x, x \pm 1\}$. For fixed m it is not hard to find all vectors $(s_1, s_2, \ldots, s_m)$, $m \le n-3$, satisfying

$$\sum_{j=0}^{q-1} d_{p+3j} = \pi_{p+3q} - \pi_p \ne 0, \quad p, q \ge 1. \tag{5}$$

If $q = 1$ then this condition becomes $\pi_{i+3} - \pi_i = s_{i+1} - s_i \ne 0$, $1 \le i < m$. We consider the vectors $(s_1, s_2, \ldots, s_m)$ in lexicographically decreasing order, excluding those with some two subsequent components equal. Without loss of generality, suppose $s_1 = x+1$. Table I demonstrates the determination of 11 solutions $(s_1 - x, s_2 - x, \ldots, s_m - x)$ of (5) if $m = 5$. For example, the first row corresponds to the sequence $(x+1, x, x+1, x, x+1)$, where $d_1 = s_2 - s_1 = -1$, $d_4 = s_5 - s_4 = 1$, $d_1 + d_4 = 0$; hence this vector is not a solution of (5). The longer Table III in the Appendix demonstrates the determination of all 8 solutions $(s_1 - x, s_2 - x, \ldots, s_m - x)$ of (5) if $m = 15$. In Table III the values 1 and -1 of $s_i - x$ are written as $+$ and $-$, respectively.

Let $A = (0, -1, 1)^{q-2}$, $B = ((1, -1, 0)^{q-2}$, where exponential notation denotes repetition. The proof by induction that

TABLE I
Determination of 11 solutions $(s_1 - x, s_2 - x, \ldots, s_5 - x)$ of (5) for $m = 5$.

1	2	3	4	5	
1	0	1	0	1	$d_1 + d_4 = 0$
1	0	1	0	-1	solution 1
1	0	1	-1	1	solution 2
1	0	1	-1	0	$d_1 + d_4 = 0$
1	0	-1	1	0	solution 3
1	0	-1	1	-1	solution 4
1	0	-1	0	1	$d_1 + d_4 = 0$
1	0	-1	0	-1	solution 5
1	-1	1	0	1	solution 6
1	-1	1	0	-1	solution 7
1	1	1	-1	1	$d_1 + d_4 = 0$
1	-1	1	-1	0	solution 8
1	-1	0	1	0	solution 9
1	-1	0	1	-1	solution 10
1	1	0	-1	1	$d_1 + d_4 = 0$
1	-1	0	-1	0	solution 11

if $m = 3q$, then the 8 solutions of (5) are given by

$$1, 0, 1, A, 0, -1, 1 \tag{6}$$
$$1, 0, 1, A, 0, -1, 0 \tag{7}$$
$$1, A, 0, -1, 1, 0, -1 \tag{8}$$
$$1, -1, 1, A, 0, -1, 1 \tag{9}$$
$$1, -1, 1, A, 0, -1, 0 \tag{10}$$
$$1, -1, 0, B, 1, 0, 1 \tag{11}$$
$$1, -1, 0, B, 1, -1, 1 \tag{12}$$
$$1, -1, 0, B, 1, -1, 0 \tag{13}$$

is given in Table II. The 8 solutions of (5) of the length $m = 3q$ of the form (6)-(13)(starting from $m = 15$) lead to 9, 7, 8 solutions of the length $m+1$, $m+2$, $m+3$, respectively. Furthermore, all solutions of the length $m+3$ are of the form (6)-(13).

Now consider the supposedly existing permutation $\pi \in S_n$, $n = 6t+3$, such that $\mathrm{disc}(\pi, 3) = 1$. The first $n-3 = 6t$ elements $s_1 - x, s_2 - x, \ldots, s_{n-3} - x$ have one of eight forms (6)-(13), determining $d_i = s_{i+1} - s_i$, $1 \le i \le 6t-1$. Furthermore, from $\gcd(6t+3, 3) = 3$ and $\sum_{j=0}^{2t} d_{3j+1} = 0$ it follows that the value of $d_{6t+1} = -\sum_{j=0}^{2t-1} d_{3j+1}$ in eight cases (6)-(13) are $q, q, q, q+1, q+1, 2q-1, 2q, 2q$ respectively. If $q \ge 1$, then in all eight cases we have $d_{6t+1} \ge q$. Combining this with the inequality $d_{6t+1} = s_{6t+2} - s_{6t+1} \le 2$, we obtain $q \le 2$, i.e. $n \le 15$. Therefore, if $n = 6t+3 > 15$ then there is no permutation $\pi \in S_n$ such that $\mathrm{disc}(\pi, 3) = 1$, hence $\mathrm{disc}(\pi, 3) > 1$. Explicit calculation gives $\mathrm{disc}(15, 3) = 1$. □

IV. Determination of $\mathrm{msum}(2km \pm 2, 2k)$

Theorem 4. *If $m > 2$, then* $\mathrm{msum}(2km \pm 2, 2k) = 1$.

Proof. Suppose first $n = 2km+2$. Consider the permutation $\pi \in S_n$ given by

$$(\pi_{2k(i-1)+2j-1}, \pi_{2k(i-1)+2j}) = \begin{cases} ((k-j)m+i, (k+j)m-i+3), & 1 \le j \le k,\ 1 \le i \le m \\ (km+1, km+2), & j = 1,\ i = m+1 \end{cases}$$

Let $x = k(2km+3)$ be an average k-sum in π. It is easily verified that $d = ((1, -1)^{km}, -km, km)$, $s_1 = s_3 = \cdots =$

TABLE II
PROOF BY INDUCTION THAT ALL SOLUTIONS OF (5) ARE GIVEN BY (6)-(13).

	$(s_1 - x, \ldots, s_{m+3} - x)$	
m	1, 0, 1, A, 0, -1, 1,	solution m,1
m+1	1, 0, 1, A, 0, -1, 1, 0	solution m+1,1
m+2	1, 0, 1, A, 0, -1, 1, 0, 1	$d_{m-2} + d_{m+1} = 0$
m+2	1, 0, 1, A, 0, -1, 1, 0, -1	solution m+2,1
m+3	1, 0, 1, A, 0, -1, 1, 0, -1, 1	solution m+3,1
m+3	1, 0, 1, A, 0, -1, 1, 0, -1, 0	solution m+3,2
m+1	1, 0, 1, A, 0, -1, 1, -1	solution m+1,2
m+2	1, 0, 1, A, 0, -1, 1, -1, 1	$d_{m-5} + d_{m-2} + d_{m+1} = 0$
m+2	1, 0, 1, A, 0, -1, 1, -1, 0	$d_{m-2} + d_{m+1} = 0$
m	1, 0, 1, A, 0, -1, 0	solution m,2
m+1	1, 0, 1, A, 0, -1, 0, 1	$d_{m-3} + d_m = 0$
m+1	1, 0, 1, A, 0, -1, 0, -1	solution m+1,3
m+2	1, 0, 1, A, 0, -1, 0, -1, 1	$d_{m-5} + d_{m-2} + d_{m+1} = 0$
m+2	1, 0, 1, A, 0, -1, 0, -1, 0	$d_{m-2} + d_{m+1} = 0$
m	1, A, 0, -1, 1, 0, -1	solution m,3
m+1	1, A, 0, -1, 1, 0, -1, 1	solution m+1,4
m+2	1, A, 0, -1, 1, 0, -1, 1, 0	solution m+2,2
m+3	1, A, 0, -1, 1, 0, -1, 1, 0, 1	$d_{m-1} + d_{m+2} = 0$
m+3	1, A, 0, -1, 1, 0, -1, 1, 0, -1	solution m+3,3
m+2	1, A, 0, -1, 1, 0, -1, 1, -1	solution m+2,3
m+3	1, A, 0, -1, 1, 0, -1, 1, -1, 1	$d_{m-4} + d_{m-1} + d_{m+2} = 0$
m+3	1, A, 0, -1, 1, 0, -1, 1, -1, 0	$d_{m-1} + d_{m+2} = 0$
m+1	1, A, 0, -1, 1, 0, -1, 0	solution m+1,5
m+2	1, A, 0, -1, 1, 0, -1, 0, 1	$d_{m-2} + d_{m+1} = 0$
m+2	1, A, 0, -1, 1, 0, -1, 0, -1	solution m+2,4
m+3	1, A, 0, -1, 1, 0, -1, 0, -1, 1	$d_{m-4} + d_{m-1} + d_{m+2} = 0$
m+3	1, A, 0, -1, 1, 0, -1, 0, -1, 0	$d_{m-1} + d_{m+2} = 0$
m	1, A, 0, -1, 1, -1, 1	$d_8 + d_{m-4} + d_{m-1}$
m	1, -1, 1, A, 0, -1, 1	solution m,4
m+1	1, -1, 1, A, 0, -1, 1, 0	solution m+1,6
m+2	1, -1, 1, A, 0, -1, 1, 0, 1	$d_{m-2} + d_{m+1} = 0$
m+2	1, -1, 1, A, 0, -1, 1, 0, -1	solution m+2,5
m+3	1, -1, 1, A, 0, -1, 1, 0, -1, 1	solution m+3,4
m+3	1, -1, 1, A, 0, -1, 1, 0, -1, 0	solution m+3,5
m+1	1, -1, 1, A, 0, -1, 1, -1	solution m+1,7
m+2	1, -1, 1, A, 0, -1, 1, -1, 1	$d_{m-5} + d_{m-2} + d_{m+1} = 0$
m+2	1, -1, 1, A, 0, -1, 1, -1, 0	$d_{m-2} + d_{m+1} = 0$
m	1, -1, 1, A, 0, -1, 0	solution m,5
m+1	1, -1, 1, A, 0, -1, 0, 1	$d_{m-3} + d_m = 0$
m+1	1, -1, 1, A, 0, -1, 0, -1	solution m+1,8
m+2	1, -1, 1, A, 0, -1, 0, -1, 1	$d_{m-5} + d_{m-2} + d_{m+1} = 0$
m+2	1, -1, 1, A, 0, -1, 0, -1, 0	$d_{m-2} + d_{m+1} = 0$
m	1, -1, 0, B, 1, 0, 1	solution m,6
m+1	1, -1, 0, B, 1, 0, 1, 0	$d_{m-3} + d_m = 0$
m+1	1, -1, 0, B, 1, 0, 1, -1	$d_9 + d_{m-3} + d_m = 0$
m	1, -1, 0, B, 1, -1, 1	solution m,7
m+1	1, -1, 0, B, 1, -1, 1, 0	$d_{m-3} + d_m = 0$
m+1	1, -1, 0, B, 1, -1, 1, -1	$d_9 + d_{m-3} + d_m = 0$
m	1, -1, 0, B, 1, -1, 0	solution m,7
m+1	1, -1, 0, B, 1, -1, 0, 1	solution m+1,9
m+2	1, -1, 0, B, 1, -1, 0, 1, 0	solution m+2,6
m+3	1, -1, 0, B, 1, -1, 0, 1, 0, 1	solution m+3,6
m+3	1, -1, 0, B, 1, -1, 0, 1, 0, -1	$d_{m-1} + d_{m+2} = 0$
m+2	1, -1, 0, B, 1, -1, 0, 1, -1	solution m+2,7
m+3	1, -1, 0, B, 1, -1, 0, 1, -1, 1	solution m+3,7
m+3	1, -1, 0, B, 1, -1, 0, 1, -1, 0	solution m+3,8
m+1	1, -1, 0, B, 1, -1, 0, -1	$d_{m-3} + d_m = 0$

TABLE III
PROOF THAT IF $m = 15$, THEN THE 8 SOLUTIONS OF (5) ARE (6)-(13).

m	$s_1 - x, s_2 - x, \ldots, s_m - x$	comment
5	+ 0 + 0 +	$d_1 + d_4 = 0$
8	+ 0 + 0 − + 0 +	$d_4 + d_7 = 0$
11	+ 0 + 0 − + 0 − + 0 +	$d_7 + d_{10} = 0$
14	+ 0 + 0 − + 0 − + 0 − + 0 +	$d_{10} + d_{13} = 0$
15	+ 0 + 0 − + 0 − + 0 − + 0 − +	solution 15, 1
15	+ 0 + 0 − + 0 − + 0 − + 0 − 0	solution 15, 2
14	+ 0 + 0 − + 0 − + 0 − + − +	$d_7 + d_{10} + d_{13} = 0$
14	+ 0 + 0 − + 0 − + 0 − + − 0	$d_{10} + d_{13} = 0$
13	+ 0 + 0 − + 0 − + 0 − 0 +	$d_9 + d_{12} = 0$

$s_{2km} = x - 1/2$, $s_2 = s_4 = \cdots = s_{2km-1} = x + 1/2$, and $s_{2km+2} = x - km - 1/2$, hence $\mathrm{msum}(\pi, 2k) = 1$.

Suppose now $n = 2km - 2$. Let the permutation $\pi \in S_n$ is defined by

$$(\pi_{2k(i-1)+2j-1}, \pi_{2k(i-1)+2j} = \begin{cases} ((i, 2km - i - 1), & j = k,\ 1 \le i \le m \\ (km - 1, km), & j = k - 1,\ i = m \\ (jm + i - 1, (2k - j)m - i), & \text{otherwise} \end{cases}$$

The average k-sum in π is now $x = k(2km - 1)$. The vector d is $((1, -1)^{km-2}, -km + 2, km - 2)$, implying $s_1 = s_3 = \cdots = s_{2km-3} = x - 1/2$, $s_2 = s_4 = \cdots = s_{2km-4} = x + 1/2$, $s_{2km+2} = x - km + 3/2$ and $\mathrm{disc}(\pi, 2k) = 1$. □

From [1][Theorem 9] it follows $\mathrm{disc}(2kt \pm 2, 2k) \le 2$, while by Theorem 4 $\mathrm{msum}(2kt \pm 2, 2k) = 1$. In many cases $\mathrm{disc}(2kt \pm 2, 2k)$ is strictly greater than $\mathrm{msum}(2kt \pm 2, 2k)$. Explicit computation in [2] by backtracking algorithm similar to the one from [4] shows that $\mathrm{disc}(4t + 2, 4) = 2$ if $10 \le 4t+2 \le 54$; $\mathrm{disc}(6t \pm 2, 6) = 2$ if $10 \le n = 6t \pm 2 \le 38$; $\mathrm{disc}(8t \pm 2, 8) = 2$ if $14 \le n = 6t \pm 2 \le 30$.

APPENDIX

Proof that if $m = 15$, then the 8 solutions of (5) are (6)-(13) is given in Table III.

REFERENCES

[1] Anstee R., Ferguson R., Griggs J. R., "Permutations with Low Discrepancy Consecutive k-sums," *Journal of Combinatorial Theory. Seriaes A*, **100**, 2002, pp. 302-321

[2] Stefanović M., "Permutations with small variations of subsequent k-sums," Master thesis, Faculty of Mathematics, Belgrade University, 2010.

[3] Rolfe T. J., "Backtracking Algorithms," Dr. Dobb's Journal, **29**, No. 5, 2004, pp. 48, 50–51

[4] Rolfe T. J., Purdom P. W., "An Alternative Problem for Backtracking and Bounding," Inroads (bulletin of the ACM SIG on Computer Science Education), **36**, No. 4, 2004, pp. 83-84.

[5] Freud, R., "On Sums of Subsequent Terms of Permutations," Acta Math. Hung. **41** No. 1–2, 1983, pp. 178–185.

[6] Erdös, P., Freud, R., Hegyvári, N., "Arithmetical properties of permutations of integers," Acta Math. Hung. **41** No. 1–2, 1983, pp. 169–176.

[7] Liu, C. L., "Elements of Discrete Mathematics," McGrawHill Computer Science Series, McGrawHill Book Co., New York, 1977.

[8] Bélisle, C., "Le Probléme du Coureur et son Interprétation Probabiliste," Ann. Sci. Math. Québec **19** 1995, pp. 1-8.

[9] Morris K., "An analysis of K-sums on N-wheels," Masters Thesis, University of South Carolina, 2000.

[10] The ACM-ICPC International Collegiate Programming Contest 2002/2003, "2608 - Constrained Circular Permutations," https://icpcarchive.ecs.baylor.edu/external/26/2608.pdf

TABLE III
(CONTINUED)

m	$s_1 - x, s_2 - x, \ldots, s_m - x$	comment
14	+ 0 + 0 − + 0 − + 0 − 0 − +	$d_7 + d_{10} + d_{13} = 0$
14	+ 0 + 0 − + 0 − + 0 − 0 − 0	$d_{10} + d_{13} = 0$
11	+ 0 + 0 − + 0 − + − +	$d_4 + d_7 + d_{10} = 0$
11	+ 0 + 0 − + 0 − + − 0	$d_7 + d_{10} = 0$
10	+ 0 + 0 − + 0 − 0 +	$d_6 + d_9 = 0$
11	+ 0 + 0 − + 0 − 0 − +	$d_4 + d_7 + d_{10} = 0$
11	+ 0 + 0 − + 0 − 0 − 0	$d_7 + d_{10} = 0$
8	+ 0 + 0 − + − +	$d_1 + d_4 + d_7 = 0$
8	+ 0 + 0 − + − 0	$d_4 + d_7 = 0$
7	+ 0 + 0 − 0 +	$d_3 + d_6 = 0$
8	+ 0 + 0 − 0 − +	$d_1 + d_4 + d_7 = 0$
8	+ 0 + 0 − 0 − 0	$d_4 + d_7 = 0$
6	+ 0 + − + 0	$d_2 + d_5 = 0$
7	+ 0 + − + − +	$d_3 + d_6 = 0$
10	+ 0 + − + − 0 + 0 +	$d_3 + d_6 + d_9 = 0$
10	+ 0 + − + − 0 + 0 −	$d_6 + d_9 = 0$
11	+ 0 + − + − 0 + − + 0	$d_7 + d_{10} = 0$
11	+ 0 + − + − 0 + − + −	$d_1 + d_4 + d_7 + d_{10} = 0$
10	+ 0 + − + − 0 + − 0	$d_3 + d_6 + d_9 = 0$
8	+ 0 + − + − 0 −	$d_1 + d_4 + d_7 = 0$
5	+ 0 + − 0	$d_1 + d_4 = 0$
6	+ 0 − + 0 +	$d_2 + d_5 = 0$
9	+ 0 − + 0 − + 0 +	$d_5 + d_8 = 0$
12	+ 0 − + 0 − + 0 − + 0 +	$d_8 + d_{11} = 0$
15	+ 0 − + 0 − + 0 − + 0 − + 0 +	$d_{11} + d_{14} = 0$
15	+ 0 − + 0 − + 0 − + 0 − + 0 −	solution 15, 3
15	+ 0 − + 0 − + 0 − + 0 − + − +	$d_8 + d_{11} + d_{14} = 0$
15	+ 0 − + 0 − + 0 − + 0 − + − 0	$d_{11} + d_{14} = 0$
14	+ 0 − + 0 − + 0 − + 0 − 0 +	$d_{10} + d_{13} = 0$
15	+ 0 − + 0 − + 0 − + 0 − 0 − +	$d_8 + d_{11} + d_{14} = 0$
15	+ 0 − + 0 − + 0 − + 0 − 0 − 0	$d_{11} + d_{14} = 0$
12	+ 0 − + 0 − + 0 − + − +	$d_5 + d_8 + d_{11} = 0$
12	+ 0 − + 0 − + 0 − + − 0	$d_8 + d_{11} = 0$
11	+ 0 − + 0 − + 0 − 0 +	$d_7 + d_{10} = 0$
12	+ 0 − + 0 − + 0 − 0 − +	$d_5 + d_8 + d_{11} = 0$
12	+ 0 − + 0 − + 0 − 0 − 0	$d_8 + d_{11} = 0$
9	+ 0 − + 0 − + − +	$d_2 + d_5 + d_8 = 0$
9	+ 0 − + 0 − + − 0	$d_5 + d_8 = 0$
8	+ 0 − + 0 − 0 +	$d_4 + d_7 = 0$
9	+ 0 − + 0 − 0 − +	$d_2 + d_5 + d_8 = 0$
9	+ 0 − + 0 − 0 − 0	$d_5 + d_8 = 0$
9	+ 0 − + − + 0 + 0	$d_2 + d_5 + d_8 = 0$
9	+ 0 − + − + 0 + −	$d_5 + d_8 = 0$
10	+ 0 − + − + 0 − + 0	$d_3 + d_6 + d_9 = 0$
13	+ 0 − + − + 0 − + − + 0 +	$d_3 + d_6 + d_9 + d_{12} = 0$
14	+ 0 − + − + 0 − + − + 0 − +	$d_1 + d_4 + \ldots + d_{13} = 0$
14	+ 0 − + − + 0 − + − + 0 − 0	$d_4 + d_7 + d_{10} + d_{13} = 0$
12	+ 0 − + − + 0 − + − + −	$d_8 + d_{11} = 0$
11	+ 0 − + − + 0 − + − 0	$d_7 + d_{10} = 0$
10	+ 0 − + − + 0 − 0 +	$d_6 + d_9 = 0$
10	+ 0 − + − + 0 − 0 −	$d_3 + d_6 + d_9 = 0$
7	+ 0 − + − + −	$d_3 + d_6 = 0$
6	+ 0 − + − 0	$d_2 + d_5 = 0$
5	+ 0 − 0 +	$d_1 + d_4 = 0$
7	+ 0 − 0 − + 0	$d_3 + d_6 = 0$
8	+ 0 − 0 − + − +	$d_1 + d_4 + d_7 = 0$
8	+ 0 − 0 − + − 0	$d_4 + d_7 = 0$
6	+ 0 − 0 − 0	$d_2 + d_5 = 0$
7	+ − + 0 + 0 +	$d_3 + d_6 = 0$
9	+ − + 0 + 0 − + 0	$d_2 + d_5 + d_8 = 0$
10	+ − + 0 + 0 − + − +	$d_3 + d_6 + d_9 = 0$
10	+ − + 0 + 0 − + − 0	$d_6 + d_9 = 0$
8	+ − + 0 + 0 − 0	$d_1 + d_4 + d_7 = 0$
6	+ − + 0 + −	$d_2 + d_5 = 0$
8	+ − + 0 − + 0 +	$d_4 + d_7 = 0$
11	+ − + 0 − + 0 − + 0 +	$d_7 + d_{10} = 0$
14	+ − + 0 − + 0 − + 0 − + 0 +	$d_{10} + d_{13} = 0$
15	+ − + 0 − + 0 − + 0 − + 0 − +	solution 15, 4
15	+ − + 0 − + 0 − + 0 − + 0 − 0	solution 15, 5
14	+ − + 0 − + 0 − + 0 − + − +	$d_7 + d_{10} + d_{13} = 0$
14	+ − + 0 − + 0 − + 0 − + − 0	$d_{10} + d_{13} = 0$
13	+ − + 0 − + 0 − + 0 − 0 +	$d_9 + d_{12} = 0$
14	+ − + 0 − + 0 − + 0 − 0 − +	$d_7 + d_{10} + d_{13} = 0$
14	+ − + 0 − + 0 − + 0 − 0 − 0	$d_{10} + d_{13} = 0$

TABLE III
(CONTINUED)

m	$s_1 - x, s_2 - x, \ldots, s_m - x$	comment
11	+ − + 0 − + 0 − + − +	$d_4 + d_7 + d_{10} = 0$
11	+ − + 0 − + 0 − + − 0	$d_7 + d_{10} = 0$
10	+ − + 0 − + 0 − 0 +	$d_6 + d_9 = 0$
11	+ − + 0 − + 0 − 0 − +	$d_4 + d_7 + d_{10} = 0$
11	+ − + 0 − + 0 − 0 − 0	$d_7 + d_{10} = 0$
11	+ − + 0 − + − + 0 + 0	$d_4 + d_7 + d_{10} = 0$
11	+ − + 0 − + − + 0 + −	$d_7 + d_{10} = 0$
12	+ − + 0 − + − + 0 − + 0	$d_5 + d_8 + d_{11} = 0$
14	+ − + 0 − + − + 0 − + − + 0	$d_1 + d_4 + \ldots + d_{13} = 0$
14	+ − + 0 − + − + 0 − + − + −	$d_{10} + d_{13} = 0$
13	+ − + 0 − + − + 0 − + − 0	$d_9 + d_{12} = 0$
11	+ − + 0 − + − + 0 − 0	$d_1 + d_4 + d_7 + d_{10} = 0$
9	+ − + 0 − + − + −	$d_5 + d_8 = 0$
8	+ − + 0 − + − 0	$d_4 + d_7 = 0$
7	+ − + 0 − 0 +	$d_3 + d_6 = 0$
9	+ − + 0 − 0 − + 0	$d_5 + d_8 = 0$
10	+ − + 0 − 0 − + − +	$d_3 + d_6 + d_9 = 0$
10	+ − + 0 − 0 − + − 0	$d_6 + d_9 = 0$
8	+ − + 0 − 0 − 0	$d_4 + d_7 = 0$
5	+ − + − +	$d_1 + d_4 = 0$
8	+ − + − 0 + 0 +	$d_1 + d_4 + d_7 = 0$
8	+ − + − 0 + 0 −	$d_4 + d_7 = 0$
9	+ − + − 0 + − + 0	$d_5 + d_8 = 0$
10	+ − + − 0 + − + − +	$d_6 + d_9 = 0$
12	+ − + − 0 + − + − 0 + 0	$d_2 + d_5 + d_8 + d_{11} = 0$
14	+ − + − 0 + − + − 0 + − + 0	$d_{10} + d_{13} = 0$
14	+ − + − 0 + − + − 0 + − + −	$d_1 + d_4 + \ldots + d_{13} = 0$
13	+ − + − 0 + − + − 0 + − 0	$d_6 + d_9 + d_{12} = 0$
11	+ − + − 0 + − + − 0 −	$d_1 + d_4 + d_7 + d_{10} = 0$
8	+ − + − 0 + − 0	$d_1 + d_4 + d_7 = 0$
7	+ − + − 0 − +	$d_3 + d_6 = 0$
8	+ − + − 0 − 0 +	$d_1 + d_4 + d_7 = 0$
8	+ − + − 0 − 0 −	$d_4 + d_7 = 0$
7	+ − 0 + 0 + 0	$d_3 + d_6 = 0$
9	+ − 0 + 0 + − + 0	$d_5 + d_8 = 0$
9	+ − 0 + 0 + − + −	$d_2 + d_5 + d_8 = 0$
8	+ − 0 + 0 + − 0	$d_4 + d_7 = 0$
6	+ − 0 + 0 −	$d_2 + d_5 = 0$
7	+ − 0 + − + 0	$d_3 + d_6 = 0$
8	+ − 0 + − + − +	$d_4 + d_7 = 0$
11	+ − 0 + − + − 0 + 0 +	$d_4 + d_7 + d_{10} = 0$
11	+ − 0 + − + − 0 + 0 −	$d_7 + d_{10} = 0$
12	+ − 0 + − + − 0 + − + 0	$d_8 + d_{11} = 0$
13	+ − 0 + − + − 0 + − + − +	$d_9 + d_{12} = 0$
14	+ − 0 + − + − 0 + − + − 0 +	$d_1 + d_4 + \ldots + d_{13} = 0$
14	+ − 0 + − + − 0 + − + − 0 −	$d_4 + d_7 + d_{10} + d_{13} = 0$
11	+ − 0 + − + − 0 + − 0	$d_4 + d_7 + d_{10} = 0$
10	+ − 0 + − + − 0 − +	$d_6 + d_9 = 0$
10	+ − 0 + − + − 0 − 0	$d_3 + d_6 + d_9 = 0$
10	+ − 0 + − 0 + 0 + 0	$d_6 + d_9 = 0$
10	+ − 0 + − 0 + 0 + −	$d_3 + d_6 + d_9 = 0$
9	+ − 0 + − 0 + 0 −	$d_5 + d_8 = 0$
10	+ − 0 + − 0 + − + 0	$d_6 + d_9 = 0$
10	+ − 0 + − 0 + − + −	$d_3 + d_6 + d_9 = 0$
13	+ − 0 + − 0 + − 0 + 0 + 0	$d_9 + d_{12} = 0$
13	+ − 0 + − 0 + − 0 + 0 + −	$d_6 + d_9 + d_{12} = 0$
12	+ − 0 + − 0 + − 0 + 0 −	$d_8 + d_{11} = 0$
13	+ − 0 + − 0 + − 0 + − + 0	$d_9 + d_{12} = 0$
13	+ − 0 + − 0 + − 0 + − + −	$d_6 + d_9 + d_{12} = 0$
15	+ − 0 + − 0 + − 0 + − 0 + 0 +	solution 15, 6
15	+ − 0 + − 0 + − 0 + − 0 + 0 −	$d_{11} + d_{14} = 0$
15	+ − 0 + − 0 + − 0 + − 0 + − +	solution 15, 7
15	+ − 0 + − 0 + − 0 + − 0 + − 0	solution 15, 8
13	+ − 0 + − 0 + − 0 + − 0 −	$d_9 + d_{12} = 0$
10	+ − 0 + − 0 + − 0 −	$d_6 + d_9 = 0$
7	+ − 0 + − 0 −	$d_3 + d_6 = 0$
5	+ − 0 − +	$d_1 + d_4 = 0$
8	+ − 0 − 0 + 0 +	$d_1 + d_4 + d_7 = 0$
8	+ − 0 − 0 + 0 −	$d_4 + d_7 = 0$
9	+ − 0 − 0 + − + 0	$d_5 + d_8 = 0$
9	+ − 0 − 0 + − + −	$d_2 + d_5 + d_8 = 0$
8	+ − 0 − 0 + − 0	$d_1 + d_4 + d_7 = 0$
6	+ − 0 − 0 −	$d_2 + d_5 = 0$

Cities, People, and Adapted Small Urban Space

The Geography of Real Space, Used in Real Time, by Real People

Cotner, Douglas

Abstract - This work advances a theory of people, spacer, and their interconnections in small space settings within cities, towns, and villages. This theory, "A Unified Field Theory of Adapted Space" advances a new language with which to study humans and their respective environments. The work herein advances the concept of "Adapted Space" in the study of the nature, distribution, and interconnectedness of the many small urban/human spaces that make up the interstitial fabric of the Human Ecology the comprises the Built-Environment."

The details of the suggested theory include the factors of the Catalytic-Interface, Spatial Scale, Physical Thresholds, Pedestrian Oases, Spatial Carrying Capacity, Accepted but Unwritten Rules of Behavior, Natural Elements, and Connecting Place with the Experience of Place. These factors are therefore critical to theoretical structure and explain the theory advanced herein.

It is therefore the hope of this work to advance a working theory that addresses people and their environments that will be revelatory and not merely derivative. To this end, the research for this theoretical development has taken place in North America, Southern Mexico, and Central Asia, where underlying concepts were tested.

Key Words - Geography, Adapted Space, Unified Field Theory

1. INTRODUCTION

I came to this work during the course of my Master's, and then later, during my Doctoral Studies program. Both my Master's Thesis ("The Geography of Adapted Space: The Human Ecology of Spatial Adaptation," Unpublished 1995), and Doctoral Dissertation ("The Geography of Adapted Space and the Human Ecology of Sustainable Development," Unpublished 1998) were instrumental in the development of "A Unified Field Theory of Adapted Space," In these early works, I sought a unifying theory, a plausible explanation, if you will, that would reveal how real space is used and modified in real time by real people in their myriad choices to meet their daily human needs, within a variety of small urban space environments. Subsequently, this study would continue on a postdoctoral basis from 1998 onwards.

The first peer-reviewed work in the quest for a unifying theory would take place in the summer of 2005. The paper, "The Urban Ecological Entropic Black Hole: A Measured Discourse Concerning the Mega-City, Within the Context of Both a Contemporary and Future Sustainable World" was presented at the IPSI international science an education conference at Sveti Stefan, Montenegro, under the aegis of the University of Belgrade. This paper elicited a considerable response from the 250 scientists, educators, urban planners, and policy expert who were then in attendance.

Since 2005, research on this theory has continued apace. "A Unified Field Theory of Adapted Space" began to emerge as the unifying theory that would tease out the heretofore poorly understood dynamics of spatial adaption of small space environments in the intimate urban environments with which environmental-users must daily contend.

This paradigm changing theory states that, "People and aggregations of people, sometimes called institutions, adapt/ modify space for the functional purposes of habitation, work, play, repose, social interaction, and acquisitive needs, sometimes temporal and sometimes transcendental." [1]

A Unified Field Theory of Adapted Space recognizes implicitly, that "Adapted Space" addresses human interaction and behaviors, in terms of discrete space that is adapted for particular human purposes, such as habitation, works and play, to cite three such purposes.

2. A UNIFIED FIELD THEORY OF ADAPTED SPACE

A Unified Field Theory of Adapted Space is a recombinant theoretical construct. It seeks to unify both the micro and macro scales of human settlement and activity. This construct

Manuscript submitted May 2014.
Dr. Douglas Cotner is with The Hemispheric Research Laboratory and Policy Institute (e-mail: doctorcotner@earthlink.net)

[1] Cotner, Douglas M., Unpublished Dissertation, 1998.

examines afresh, the antecedent geographical and urban Planning ideas of “Sequent Occupance” (Whittlesey, 1929, and Meyer 1935), “Landscape Ecology” (Barrows, 1922), “Culture History” (Sauer, 1925), and the “Spatial Adaptation Behavior (Whyte, 1980) of humans, as human and cultural modifiers of the humanized environments of place and environment. This construct also seeks to cognitively capture, and make intellectually apprehendable, the impact of cultural inflections of people on discrete places (people acting on space and space acting on people), within a regional context, resulting in a unified paradigm.

A Unified Field Theory of Adapted Space underpins the Ecological Footprints of people and their consequent importance for sustainability of human and other communities, as interacting elements within the Biosphere and Geosphere. Ecological footprints, which are functions of the consumption of biophysical goods, and which represent the dynamics of how, why, and the means by which people adapt or modify real space on Earth to meet their physical, emotional, and intellectual needs. These needs lead to the consumption of many things; among then are the basics of food, fuel, and fiber. Thus, an understanding how real people adapt real space in real time begins to emerge. Ultimately, A Unified Field Theory of Adapted Space suggests that we must go beyond just the Ecological Footprint if there is to be a successful understanding of complex mainland interactions. Thus, this analysis leads to solutions of the great human and associated environmental problems that are now confronting us. These problems include those of population (its growth and distribution), natural resources adequacy, livable living space, transportation, clean air and water, genetically modified foods and sustainable energy availability.

The notion of adapted space and the theory to support such an idea, evolved over a number of years from 1989 onwards. The goal of this research has been to explore and test such an idea. The Field work for this research in the United States and Southern Mexico principally. The outcome of this work endeavored to answer the basic questions concerning how and why people (environmental users) actually modified space for personal and collective purposes, and at what scale. Antecedently, the work of William H. Whyte was most helpful in the visual documentation of the actions of real people in real time, adapting and modifying real space for both individual and group purposes to meet specific user needs. Before Whyte’s seminal work “The Social Life of Small Urban Spaces” that was published in the year 1980, an understanding of urban adaptation was not widely understood. There was some related work, however, done by American Geographers, but their approach was overall, at the macro scale, and merely additive. Since the time of Whyte’s work, very little new research to extend his findings seems to have emerged. Consequently, A Unified Field Theory of Adapted Space represents possibly, the only new approach by which to understand the newest of urban phenomena, the “Urban Ecological Entropic Black Hole” or the Mega-City. Consequently, when addressing this new human settlement type, one rapidly realizes that only a small portion of the work of Geographers and Urban Planners, carefully built up over the past 150 years, is of any use whatsoever, when the Mega-City is subjected to analysis, especially when problems of ecologically sustainability are considered. For these complex problems,

A Unified Field Theory of Adapted Space is both relevant and important. Thus, the Mega-City is a revolutionary new human settlement type that lies beyond the existing analytical models of Geographers, Urban Planners, and Public Policy Makers.

3. ANTECENDENTS OF THIS WORK

Though the utilization of an expressly formalized *Systems Theory* has only relatively recently been introduced into the Field of Geography, the history and use of systematic thinking in Geography is closely associated with *Functional Analysis.* This viewpoint regards regions as complex entities, which is to say, *Geography as Human Ecology*. Elements such as systematic thinking may be found in the works of Ritter, Vidal de la Blache, Brunhes, Whittlesey, and Sauer, to cite the more notable in this regard (Holt-Jensen, Geography: Its History and Concepts, 1980).

It was during the 1970s, however, that such a Systems Theory itself emerged into clearer focus. Thus, in a similar fashion, as one finds, as one find in an appreciation for the logic of Functional Analysis, which has led towards a more formalized Systems Concept, one of the mainstreams of contemporary geographical inquiry, we have consequently witnessed a move in the direction of systematic thinking in the field. W now seem to regard this intellectual construct as being of critical importance, because of the possibility that it presents of uniting Geographical Methods with Geographical Philosophy, (Holt-Jensen). The most obvious and recognizable antecedent of systematic thinking was the school of thought, which, at the beginning of the twentieth century, attempted to define Geography as *Human Ecology*, (Holt-Jensen, 1980). The ecological concepts that we seem to find implicit in the works of Vidal de la Blache are somewhat different from those spoken by Harlan Barrows in his presidential address to the Association of American Geographers in 1922, (Barrows, 1922).

In his now historic address, he opined that the practice of Geography should concentrate on the Study of Man’s associations with the natural environment. He argues here, that the subject of

Geography had overly specialized; what we would today call, *fissiparous* tendencies. Furthermore, such specialized branches as Geomorphology, Climatology, and Biogeography should be excised from the core of the subject. Moreover, that Geography should focus on those realms that would lead in the direction of a synthesis, with an Economic Regional Geography occupying center stage. Beyond this, he seems to have given no clear directions about such things as methods and research, and because Natural Science Geographers exerted a powerful influence on the practice of Geography at that time, any research concerning Geography as Human Ecology received correspondingly scant attention in the evolution of the Field. Such circumstances as these, therefore, accounted for Barrow's ideas having little impact in the 1920s. Later on, Barrow's paper "Geography as Human Ecology" was often cited as the subject that Geography should be concerned about.

The concept of "Human Ecology" may need further clarification at this point. Accordingly, Ernst Haekel, the German Biologist and popular Philosopher first used the term, "Ecology," from the Greek *Oikos* meaning home place, and *Logos* meaning doctrines in 1868. Biologists study the relationships and interactions between plants and animals, and their niches. Ecologists should, from this definition study natural relationships, whereby particular species of plants and animals are dependent on each other, as well as the nonorganic environment (soil, climate, along with the chemical composition of atmosphere and water, etc.). The aspect of Ecology that impinges upon Man's biological nature, in terms of the interaction of those aspects with his relationship to the natural environment should, therefore, constitute the subject matter of Human Ecology, and by extension, Sustainable Development.

During recent decades, however, Ecology has gained a serious new interest with the Biological Research Community, and has become the spur to the general debate about the protection of Nature, in its broadest, ecumenical sense. The marked advances in research, which have been recently achieved by Biologists, are principally attributable to the development of a new direction within Ecology, which is to say, *Systems Ecology*, which studies the construction and function of ecosystems, (Holt-Jansen, 1980).

3.1 A Useful Concept

By utilizing this contextual milieu, the concept of *Adapted Space* can be a useful one for the Geographer who is intrigue by the nature, distribution, and interconnectedness of the many small urban/human spaces that make up the interstitial fabric of the Human Ecology, comprises the *Built-Environment*. For it is within and between these small space (the daily ecological context) that the daily regime of human activities, which are associated with the many acts of social interaction, transpire. Thus, the resultant network of these spaces, represent a landscape, which offers a more insightful understanding of the human and cultural aspects that constitute the "Human Ecology of the Urban Environment", as it has come to be understood, during the final quarter of the twentieth century.

Adapted Space possesses the two fundamental elements that appeal to the mind of the inquiring Geographer, which is that of space and people, in terms of the dynamics of the resultant interaction. This Interaction has been carefully observed, and documented in recent years by William H. Whyte, (The Social Life of Small Urban Spaces, 1980). Whyte observes that small urban space either works or it does not work, in terms of human purposes.

4. DETAILS OF THEORY

4.1 Small Urban/Human Spaces Have Social Lives Too

As previously stated, in "The Social Life of Small Urban Spaces" (Whyte), either as observed via time-lapse photography, these spaces work or the not work, in terms of meeting human purposes. Such purposes, as herein set forth are *Social Interaction, Sustenance Activities, Repose, Mental Diversion, Information Acquisition, Other Acquisitive Activities, and Aesthetic Needs*. The following are therefore basic elements, which contribute to the workability of a small urban space, (The Social Life of Small Urban Spaces):

1. A Catalytic Interface—Whyte identified this as *Triangulation*. It is a condition that seems to facilitate the interaction of two or more people, in terms of an external stimulus that creates a linkage, causing strangers to engage in conversation with one another, as though they were perfectly well acquainted. An excellent example of this *interface* is the baseball game. Because of the context that the game creates, strangers become acquainted, as the result of the stimulus that the game brings. Thus, they converse almost intimately. They share a beer, eat hotdogs, and display emotion as the consequence of the catalytic action that the game brings to the venue, within which these individuals temporarily interact.

2. Spatial Scale—How wide, how deep, how expansive, and how compressive is it?

3. Physical Thresholds—These physical and mental artifacts queue and provide important visual signals, as an individual moves within and about a particular small urban space environment.

4. Pedestrian Oases—Both Inside and Outside.
   a. Sitting amenities.
   b. Food and drinking water availability.
   c. Toilets.
   d. Aesthetics.

1. Water Features
2. Light, shade, warmth, sound, feel.
3. Visual interest elements Signage, Graphics, and Sculpture.

5. Spatial Carrying Capacity; Within a volumetric unit of space and pursuant to the spatial purpose and function of a space; the number of people that can be determined. We shall later learn how this is an important factor in scientifically based Sustainable Development, but with a slightly different inflection to conventional thinking.

6. Accepted but Unwritten Rules of Behavior. When people use a particular small space for a specific purpose or group of purposes, they seem to comply with certain rules of behavior that they have learned over time, concerning what will be tolerated among various users. Kevin Lynch called this situation "*A Behavior Setting"* (Site Planning, 1989).

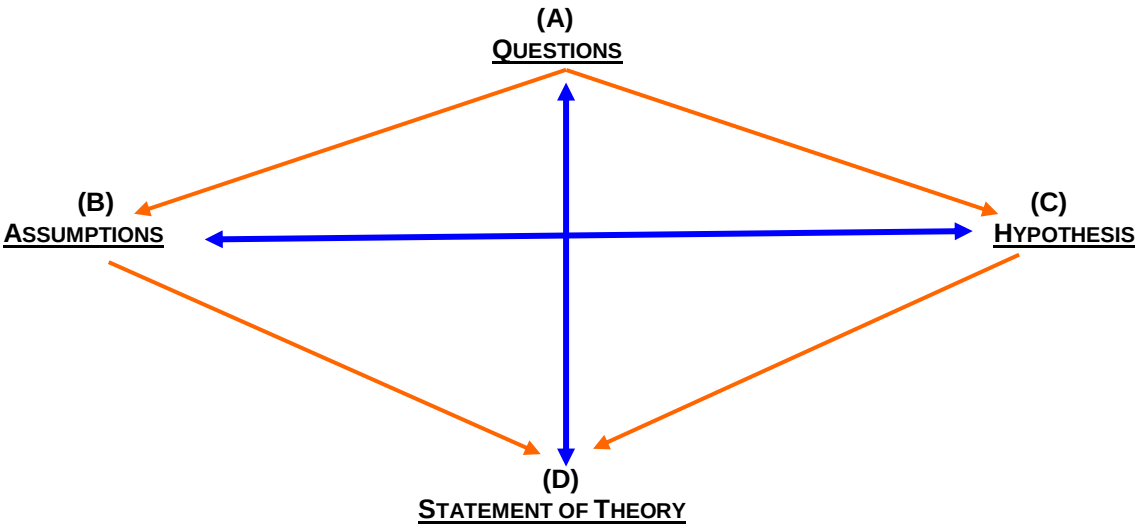
7. Natural Elements.
   1. Trees.
   2. Water.
   3. Grass.
   4. Flowers.
   5. Views of sky and clouds.

8. Connecting Place with the Experience of Place. This represents an experiential process and a sense of wholeness is created which is then perceived by the Environmental-User. Therefore, when environmental-users begin to consciously articulate that which they area apprehending on a regular basis, in terms of the daily use of a place, a subtle but profound connection is made. Place is then connected with the experience of a place through the formation of a perceptual link, as expressed by Hiss, (The Experience of Place, 1991).

The conformity of these elements to the reality of numerous *Environmental Behavior Setting*, as demonstrated by Whyte, (The Social Life of Small Urban Spaces) were determined to be *Vital Cofactors* for the success of a small urban space, along with their attendant human interaction, are numerically dependent. This is to say, that if these spaces are to succeed, there must be a sufficient midday population. He further indicates that spatial viability, within the context of this discussion is dependent upon a midday pedestrian population of not less than 1000 persons per hour in large cities and scaled accordingly as urban settlements decrease in relative size. Therefore, to generate this minimum pedestrian flow, the network of small urban space and pedestrian realms of streets that connect them, must be inviting, appearing friendly and safe, so that regular and sufficient use will occur, and thus assuring their long term Sustainability.

Conceptualizing the preceding discourse requires a thought-model. This schema clarifies densely complex spatial problems, and stimulates insight into the problem, as set forth in this paper. The likely model as herein suggested, is important because it raises more intriguing questions, just as the solution to a good mathematics question, and inevitably generates more mathematics. This thought-model is designed to do just that.

4.2 A Unified Field Theory of Adapted Space: A Likely Model



Questions Demanded of This Model:

1. How do individuals, groups of individuals, and institutions modify and adapt space at varying spatial scales over time, for their respective needs and purposes?
2. What is Adapted Space?
3. How does the adaptation of space work at varying spatial scales?
4. Can spatial adaptation reflect the intellectual and spiritual values of a society?
5. What roles do individuals and institutions play in spatial adaptation?
6. Are there spatial adaptation principles that are universally, socially, culturally, politically, and economically, relevant and applicable both First World and Developing World settings?

Assumptions:

1. Individual persons see their daily environment as a reflection of themselves, as discrete persons, within a unique relational context.
2. Institutions pursue their need to modify space in relationship to other dynamic environmental elements.

Hypothesis:

People, individually, as well as collectively, and groups of people called institutions, both public and private, operating in relative time, at variable spatial scales, modify and adapt space for diverse reasons and myriad purposes to meet their respective needs and wants.

Statement of Theory:

"A Unified Field Theory of Adapted Space" can be expressed as follows:

$$\begin{pmatrix} \textbf{(A)} \\ \textbf{[(AS) = (}\Sigma_{\textbf{ASF}}\textbf{)}} \end{pmatrix} + \begin{pmatrix} \textbf{(B) = UFTAS} \\ \textbf{(ef =}\Sigma_{\textbf{aai}}\textbf{)]} \\ \textbf{i = (1 to n)} \end{pmatrix}$$

Where:

AS = Adapted Space
∑ = Summation
ASF = Adapted Space Factors
ef = Per-Capita Ecological Footprint
aai = Ecosystem Areas Appropriated
i = Major Consumption Item
n = All Items Purchases (n) in an Annual Shopping Basket of Consumption Goods and Services

It should be noted that he equation "A Unified Field Theory of Adapted Space," Part A of this equational expression is the original work of Professor Dr. Douglas M. Cotner, as developed in his Doctoral Dissertation, at the American Institute of Urban and Regional Affairs—Unpublished, 1998, while Part B represents the work of professors William Rees and Mathis Wackernagel at the University of British Columbia in their book, "Our Ecological Footprint: Reducing Human Impact on Earth," published by New Society Publishers in 1996, as part of "The New Catalyst Bioregional Series." These two theoretical constructs are recombinant, reconsidering and thus rendering a new equational construct in support of a "Unified Field Theory of Adapted Space."

5. CONCLUDING REMARKS

To the degree that we can understand the relationship between people, their environments, and the effect of space on human behavior, and human behavior on the small urban spaces and the connections between them will in large measure, the livability of urban environments. Therefore, it is to this end that this work has been directed.

REFERENCES

1. Cotner, Douglas M., "Site Planning: A Contemporary Public Imperative," *The Geographical Survey*, Vol. 2, No. 1, (January 1975: 1-22).
2. Cotner, Douglas M., "The Site Plan: Fundamental Unit of Structure in the Integration of Man with His Environment," *The Geographical Survey*, Vol. 4, No. 3, (July 1975: 119-127).
3. Hiss, Ton, "The Experience of Place: A New Way of Looking at and Dealing with Our Radically Changing Cities and Countryside," *New York: Dover Publications*, Inc. 1991.
4. Lynch, Kevin, "Good City Form," *The MIT Press,* Cambridge, Massachusetts, 1984.
5. Lynch, Kevin, "The Image of the City," *The MIT Press,* Cambridge, Massachusetts,
6. Lynch, Kevin 1960 and Gary Hack 1983, "Site Planning 3rd edition," *The MIT Press,* Cambridge, Massachusetts,
7. Mumford, Lewis, "The City in History: Its Origins, Its Transformations, and Its Prospects," *Harcourt Brace Jovanovich Publishers*. New York, 1961.
8. Rees, W. And Wackernagel, M, "Ecological Footprints and Appropriated Carrying Capacity: Measuring the Natural Capital requirements of the Human Economy," 1994.
9. Wackernagel, M and William Rees, "Our Ecological Footprint: Reducing Human Impact on the Earth," *New Society Publishers,* Gabriola Island, BC and Philadelphia, PA: (1995).

10. Whyte, William H., “City: Rediscovering the Center,” *Dell Publishing Group, Inc.* New York: Bantam Doubleday, 1988.

11. Whyte, William H., “The Social Life of Small Urban Spaces,” The Conservation Foundation of Washington, Ann Arbor, Michigan: D.C. 1980.

Do We Have Enough Information to Reduce Musculoskeletal Disorders?

Jemec, Jure and Đorđevič, Srđan

Abstract - Musculoskeletal disorders are one of the biggest burdens of modern humanity. Total cost of musculoskeletal disorders can reach 3,4% of gross domestic product. There are four major musculoskeletal disorders and low back pain is the biggest of them. Low back pain is nonspecific pathology and can be caused by a large number of reasons. Effective methods to diminish conditions for low back pain have not been well developed at this point and also clear indication criteria for low back pain have not yet been determined. The key question concerning low back pain is, do we have relevant information about skeletal muscles, to be able to diminish conditions for low back pain? Whatever methodology would be used it must be selective, functional, valid, repeatable, non-invasive and simple to use. The aim of this review is to evaluate Tensiomyography as potential methodology that could be used in order to find critical points of each individual to avoid injury with preventive treatment. Tensiomyography is non-invasive, selective functional muscle diagnostic method that has very high repeatability. The review showed that Tensiomyography can be used to detect critical points to diminish conditions for low back pain. Furthermore Tensiomyography can contribute to optimize the process of rehabilitation and physical recovery of patients. In the future Tensiomyography will need more clinical studies in order to further develop basic characteristic models.

Key words **- Musculoskeletal disorders, osteoarthritis, rheumatoid arthritis, osteoporosis, lower back pain, Tensiomyography, Fiber type, repeatability, validity, simplicity, critical points**

1. INTRODUCTION

Musculoskeletal disorders are one of the biggest burdens of modern humanity. There are four major musculoskeletal conditions: osteoarthritis, rheumatoid arthritis, osteoporosis and lower back pain. Osteoarthritis (lost joint cartilage that leads to pain and loss of function in the knees and hips) affects 9.6% of men and 18% of woman aged > 60 years [1]. Rheumatoid arthritis is an inflammatory condition that usually affects multiple joints. It affects 0.3–1.0% of the general population and is more prevalent among women and in developed countries [1]. Osteoporosis, which is characterized by low bone mass and micro architectural deterioration, is a major risk factor for fractures of the hip, vertebrae, and distal forearm. Hip fracture is the most detrimental fracture, being associated with 20% mortality and 50% permanent loss in function [1]. Lost ability to move is considered as one of the biggest reasons for osteoporosis. Low back pain is the most prevalent of musculoskeletal conditions; it affects nearly everyone at some point in time and about 33% of the population at any given point [1]. Disease-related costs, from a societal perspective, were measured in Canada, year 1994, using prevalence based analysis. The results showed the total cost of musculoskeletal disorders in Canada was 3.4% of the gross domestic product [2]. Gross domestic product in European Union in 2012 was €12.899 trillion and 3,4% of gross domestic product in European Union represents €438,566 billion.

Jure Jemec is CEO of TMG-BMC d.o.o., Splitska 5, 1000 Ljubljana, Slovenia, Europe, (e-mail: jure.jemec@tmg.si, web: www.tmg.si),
Srđan Đorđević is head of R&D of TMG-BMC d.o.o., Splitska 5, 1000 Ljubljana, Slovenia, Europe, (e-mail: srdjandj @tmg.si).

2. LOW BACK PAIN AND DIAGNOSIS

2.1 Low back pain

Low back pain is a considerable health problem in all developed countries and is most commonly treated in primary healthcare settings. It is usually defined as pain, muscle tension, or stiffness localized below the costal margin and above the inferior gluteal folds, with or without leg pain (sciatica). The most important symptoms of non-specific low back pain are pain and disability.

2.2 Cause of low back pain

Low back pain is not specific symptom and can be caused by a large number of reasons like ageing, losing bone strength, change of muscle tone and elasticity, muscle sprain or similar. Most of low back pain follows injury to the back, but pain may also be caused by degenerative conditions such as arthritis, disc disease or others. Latest studies indicate that individual back muscle asymmetry could be one of the main reasons for low back pain. Back muscle asymmetry might also be caused by non back muscle functional asymmetry such as Gluteus muscles, Hamstring muscles, Quadriceps muscles or facial muscles.

2.3 Prevention of low back pain

Effective methods to diminish conditions for low back pain have not been well developed [3]. Exercise is probably effective in preventing recurrences in those with pain that has lasted more than six weeks. [4] [5] There is little to no evidence that back belts are any more helpful in preventing low back pain than education about proper lifting techniques[6] [7]. Clear indication criteria for low back pain have not yet been determined.

2.4 How is low back pain diagnosed

A variety of diagnostic methods are available. *X-ray imaging* includes conventional and enhanced methods that can help diagnose the cause and site of back pain. *Computerized tomography (CT)* is a quick and painless process used when disc rupture, spinal stenosis, or damage to vertebrae is suspected as a cause of low back pain. *Magnetic resonance imaging (MRI)* is used to evaluate the lumbar region for bone degeneration or injury or pathology in tissues and nerves, muscles, ligaments, and blood vessels. *Electrodiagnostic procedures* include electromyography (EMG), nerve conduction studies, and evoked potential (EP) studies. EMG assesses the electrical activity in a nerve and can detect if muscle weakness results from injury or a problem with the nerves that control the muscles. *Bone scans* are used to diagnose and monitor infection, fracture, or disorders in the bone. Thermography involves the use of infrared sensing devices to measure small temperature changes between the two sides of the body or the temperature of a specific organ. Ultrasound imaging, also called ultrasound scanning or sonography, uses high-frequency sound waves to obtain images inside the body. Electromyography is the only selective method that was used but EMG devices don't measure muscle function and still has too high variability.

3. TENSIOMYOGRAPHY

The key question we can have concerning low back pain is do we have relevant information about skeletal muscles to be able to diminish conditions for low back pain? What kind of information about skeletal muscles would be of high importance? Whatever methodology would be used it must be selective, functional, valid, repeatable, non-invasive and simple to use. Several attempts were made, but there is no indication that selective functional muscle diagnostic would be used in order to diagnose or manage low back pain.

The aim of this review is to evaluate Tensiomyography as potential methodology that could be used in order to find critical points of each individual and to avoid injury, or if they have already occurred, to identify them with certainty and to prescribe a personalized recovery until the injury is completely recovered.

Tensiomyography (TMG) is measuring used for monitoring radial muscle belly displacement under isometric conditions. Latest studies showed that radial muscle belly displacement is of big importance since changes in the radial distance between the actin and myosin filaments, the filament lattice spacing, are responsible for between 20% and 50% of the change in force. Thus, lattice spacing is a significant force regulator, increasing the slope of muscle's force–length dependence [8].

Tensiomyography was developed in the Laboratory for bioelectromagnetics (LBM) at the Faculty of Electrical engineering, University of Ljubljana, Slovenia. The displacement sensor is positioned perpendicularly to the tangential plane on the largest area above the muscle belly. The muscle belly displacement (enlargement) during contraction is observed and monitored [9] [10].

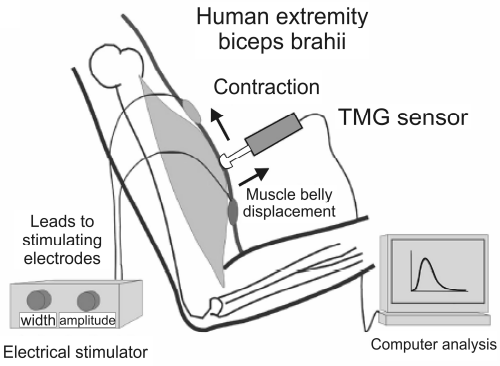


Fig. 1. Experimental setup used to evoke and measure the biceps brachii (BB) isometric twitch contraction responses. The TMG sensor measures muscle radial displacement during twitch contractions induced by short electrical stimuli. The stimulating electrodes are placed directly onto the skin.

Muscle contraction is elicited by single-twitch electrical stimuli. Two self-adhesive electrodes is placed symmetrically around the TMG sensor. The anode is placed distally and the cathode proximally, 20-50 mm from the measuring point. Bipolar ES consisted of a single DC pulse of 1 ms in duration. A typical TMG record with parameters and definitions is shown in Figure 1. The measured parameters are shown in Figure 2. These parameters are the maximal amplitude of the signal (Dm), the delay time from the stimulation to 10% of the maximal contraction (td), the time of contraction from 10% to 90% of the maximal contraction (tc), the time of sustained contraction from 50% contraction to 50% of the relaxation (ts) and the relaxation time from 10% relaxation to 50% relaxation (tr).

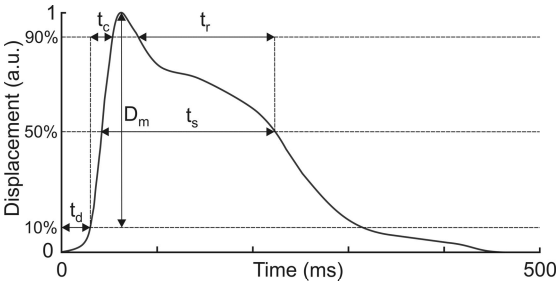


Fig. 2. (a) The parameters that were measured with the TMG signal: Dm – maximum amplitude (displacement), td – initial delay time, tc – contraction time, ts – sustained contraction time and tr – half relaxation time.

4. EVALUATION OF TENSIOMYOGRAPHY CONCERNING LOW BACK PAIN

Tensiomyography can be used as an accurate non-invasive predictor of the %MHC-I in a muscle [11] this is one of the crucial conditions for prevention of low back pain since information on muscle fiber type composition is of great importance in muscle physiology.

Tensiomyography is fast, user friendly was and repeatable diagnostic [12] [13] [14].

Tensiomyography was used to monitor pre and post operative muscles adaptation of ACL reconstruction rehabilitation process. The results demonstrate different pattern of adaptation for each skeletal muscle necessary to assure knee joint stability.

Additionally resistance to fatigue and muscle stiffness in the hamstring muscles was considered to be one of the risk factors for ACL injury in male soccer players [15]. Predominant impairment in TMG characteristics of the quadriceps over hamstrings was shown as indication of altered muscular co-contraction (imbalance) between both muscle groups, which might be another risk factor for ACL injury in this population [15]. Results of this study demonstrate the ability of Tensiomyography to provide relevant information about muscle balance around knee. Additionally muscle balance/imbalance was considered to be good indication point for ACL.

It is very important to have selective information about skeletal muscles when evaluating low back pain patients, since we need a model to explain the alterations of back muscle recruitment due to low-back pain. It is proposed that motor control changes in patients are functional in that they enhance spinal stability.

The most prevalent chronic injury in classical ballet dancers is low back pain. In classical ballet dancers, low back pain and hip-region pain are associated with a smaller cross-sectional area (CSA) of the multifidus but not the erector spinae, psoas, or quadratus lumborum muscles [16]. Magnetic resonance imaging was performed and dancers with no pain had larger multifidus muscles compared to those with low back pain. CSA can be used to estimate muscle force, but we would propose that future studies also use methodology that would be able to measure muscle contractile properties as this information would augment

It is important for tensiomyography that information can be obtained through a non-invasive method, without any effort of the subject. Regarding the low back pain, it can be even better by optimizing the process of rehabilitation to dose the load and intensity of work, or to detect local muscle fatigue as a changed ability to activate muscle fibres. Moreover, one can control the muscle properties and to detect any change in the same muscle, due to a recent injury or an inadequate amount of work demonstrated as local muscle fatigue. This information can be of great interest in the process of postoperative recovery.

5. CONCLUSION

Tensiomyography can be used to detect critical indication points to diminish conditions for low back pain. Furthermore Tensiomyography can contribute to optimize the process of rehabilitation and physical recovery of patients. A simple methodology is an important factor for the feasibility of the method. In addition, it is objective, non-invasive, rapid and selective, and can give repetitive information. The characteristics of the equipment allow assessing the data with great mobility and interpretation of results can be made automatically. Further studies are needed in order to implement Tensiomyography as a helpful tool to improve prevention and management of low back pain.

REFERENCES

[1] WOOLF, Anthony D. and PFLEGER, Bruce. Burden of major musculoskeletal conditions. *Bull World Health Organ* [online]. 2003, vol.81, n.9 [cited 2013-11-30], pp. 646-656 .

[2] Coyte, Peter C., Asche, Carl V., Croxford, Ruth and Chan, Ben, "The economic cost of musculoskeletal disorders in Canada" *Arthritis & Rheumatism*, Volume 11, Issue 5, pages 315 – 325, October 1998

[3] Hoy D, Bain C, Williams G, et al. (June 2012). "A systematic review of the global prevalence of low back pain". Arthritis Rheum. 64 (6): 2028–37.

[4] Choi BK, Verbeek JH, Tam WW, Jiang JY (2010). "Exercises for prevention of recurrences of low-back pain". In Choi, Brian KL. Cochrane Database of Systematic Reviews (1): CD006555

[5] Chou R, Qaseem A, Snow V, et al. (October 2007). "Diagnosis and treatment of low back pain: a joint clinical practice guideline from the American College of Physicians and the American Pain Society". Annals of Internal Medicine 147 (7): 478–91

[6] Guild DG (September 2012). "Mechanical therapy for low back pain". *Prim. Care* 39 (3): 511–6

[7] Sahar T, Cohen MJ, Uval-Ne'eman V, *et al.* (April 2009). "Insoles for prevention and treatment of back pain: a systematic review within the framework of the Cochrane Collaboration Back Review Group". *Spine* 34 (9): 924–33.

[8] Williams CD, Salcedo MK, Irving TC, Regnier M, Daniel TL. 2013 The length–tension curve in muscle depends on lattice spacing. Proc R Soc B 280: 20130697.

[9] Valencic, V., Knez, N., (1997). Measuring of skeletal muscles' dynamic properties. Artif Organs 21, 240-242.

[10] Dahmane R, Valencic V, Knez N, Erzen I. Evaluation of the ability non-invasive estimation of the muscle

contractile properties on the basis of the muscle belly response. J Med Biol Eng Comp 2000; 39(1): 51-5.
[11] Simunič B, Degens H, Rittweger J, et al. (2011) Noninvasive estimation of myosin heavy chain composition in human skeletal muscle. Med Sci Sports Exerc 43, 1619–1625.
[12] Križaj D., Šimunič B., Žagar T., "Short-term repeatability of parameters extracted from radial displacement of muscle belly", Journal of Electromyography and Kinesiology - August 2008 (Vol. 18, Issue 4, Pages 645-651, DOI: 10.1016/j.jelekin.2007.01.008)
[13] Tous-Fajardo J, Moras G, Rodr_ıguez-Jim_enez S, et al. (2010) Inter-rater reliability of muscle contractile property measurements using non-invasive tensiomyography. J Electromyogr Kinesiol 20, 761–766.
[14] Simunic B (2012) Between-day reliability of a method for noninvasive estimation of muscle composition. J Electromyogr Kinesiol 22, 527–530.
[15] Alentorn-Geli E, Alvarez-Diaz P, Ramon S, Marin M, Steinbacher G, Boffa JJ, Cusco X, Ballester J, Cugat R (2014) Assessment of neuromuscular risk factors for anterior cruciate ligament injury through tensiomyography in male soccer players. Knee Surg Sports Traumatol Arthrosc , DOI 10.1007/s00167-014-3018-1
[16] Gildea JE1, Hides JA, Hodges PW (2013) Size and symmetry of trunk muscles in ballet dancers with and without low back pain. J Orthop Sports Phys Ther. 2013 Aug;43(8):525-33. doi: 10.2519/jospt.2013.4523. Epub 2013 Apr 30.

Reviewers:

Australia
Abramov, Vyacheslav; Monash University
Begg, Rezaul; Victoria University
Bem, Derek; University of Western Sydney
Betts, Christopher; Pegacat Computing Pty. Ltd.
Buyya, Rajkumar; The University of Melbourne
Chapman, Judith; Australian University Limited
Chen, Yi-Ping Phoebe; Deakin University
Hammond, Mark; Flinders University
Henman, Paul; University of Queensland
Palmisano, Stephen; University of Wollongong
Ristic, Branko; Science and Technology Organisation
Sajjanhar, Atul; Deakin University
Sidhu, Amandeep; University of Technology, Sydney
Sudweeks, Fay; Murdoch University

Austria
Derntl, Michael; University of Vienna
Hug, Theo; University of Innsbruck
Loidl, Susanne; Johannes Kepler University Linz
Stockinger, Heinz; University of Vienna
Sutter, Matthias; University of Innsbruck

Brazil
Parracho, Annibal; Universidade Federal Fluminense
Traina, Agma; University of Sao Paulo
Traina, Caetano; University of Sao Paulo
Vicari, Rosa; Federal University of Rio Grande

Belgium
Huang, Ping; European Commission

Canada
Fung, Benjamin; Simon Fraser University
Grayson, Paul; York University
Gray, Bette; Alberta Education
Memmi, Daniel; UQAM
Neti, Sangeeta; University of Victoria
Nickull, Duane; Adobe Systems, Inc.
Ollivier-Gooch, Carl; The University of British Columbia
Paulin, Michele; Concordia University
Plaisent, Michel; University of Quebec
Reid, Keith; Ontario Ministry og Agriculture
Shewchenko, Nicholas; Biokinetics and Associates
Steffan, Gregory; University of Toronto
Vandenberghe, Christian; HEC Montreal

Czech Republic
Kala, Zdenek; Brno University of Technology
Korab, Vojtech; Brno University of technology
Lhotska, Lenka; Czech Technical University

Finland
Lahdelma, Risto; University of Turku
Salminen, Pekka; University of Jyvaskyla

France
Cardey, Sylviane; University of Franche-Comte
Klinger, Evelyne; LTCI – ENST, Paris
Roche, Christophe; University of Savoie
Valette, Robert; LAAS - CNRS

Germany
Accorsi, Rafael; University of Freiburg
Glatzer, Wolfgang; Goethe-University
Gradmann, Stefan; Universitat Hamburg
Groll, Andre; University of Siegen
Klamma, Ralf; RWTH Aachen University
Wurtz, Rolf P.; Ruhr-Universitat Bochum

India
Pareek, Deepak; Technology4Development
Scaria, Vinod; Institute of Integrative Biology
Shah, Mugdha; Mansukhlal Svayam

Ireland
Eisenberg, Jacob; University College Dublin

Israel
Feintuch, Uri; Hadassah-Hebrew University

Italy
Badia, Leonardo; IMT Institute for Advanced Studies
Berrittella, Maria; University of Palermo
Carpaneto, Enrico; Politecnico di Torino

Japan
Hattori, Yasunao; Shimane University
Livingston, Paisley; Linghan University
Srinivas, Hari; Global Development Research Center
Obayashi, Shigeru; Institute of Fluid Science, Tohoku University

Netherlands
Mills, Melinda C.; University of Groningen
Pires, Luís Ferreira; University of Twente

New Zealand
Anderson, Tim; Van Der Veer Institute

Portugal
Cardoso, Jorge; University of Madeira
Natividade, Eduardo; Polytechnic Institute of Coimbra
Oliveira, Eugenio; University of Porto

Singapore
Tan, Fock-Lai; Nanyang Technological University

South Korea
Kwon, Wook Hyun; Seoul National University

Spain
Barrera, Juan Pablo Soto; University of Castilla
Gonzalez, Evelio J.; University of La Laguna
Perez, Juan Mendez; Universidad de La Laguna
Royuela, Vicente; Universidad de Barcelona
Vizcaino, Aurora; University of Castilla-La Mancha
Vilarrasa, Clelia Colombo; Open University of Catalonia

Sweden
Johansson, Mats; Royal Institute of Technology

Switzerland
Niinimaki, Marko; Helsinki Institute of Physics
Pletka, Roman; AdNovum Informatik AG
Rizzotti, Sven; University of Basel
Specht, Matthias; University of Zurich

Taiwan
Lin, Hsiung Cheng; Chienkuo Technology University
Shyu, Yuh-Huei; Tamkang University
Sue, Chuan-Ching; National Cheng Kung University

United Kingdom
Ariwa, Ezendu; London Metropolitan University
Biggam, John; Glasgow Caledonian University
Coleman, Shirley; University of Newcastle
Conole, Grainne; University of Southampton
Dorfler, Viktor; Strathclyde University
Engelmann, Dirk; University of London
Eze, Emmanuel; University of Hull
Forrester, John; Stockholm Environment Institute
Jensen, Jens; STFC Rutherford Appleton Laboratory
Kolovos, Dimitrios S.; The University of York
McBurney, Peter; University of Liverpool
Vetta, Atam; Oxford Brookes University
WHYTE, William Stewart; University of Leeds
Xie, Changwen; Wicks and Wilson Limited

USA
Bach, Eric; University of Wisconsin
Bolzendahl, Catherine; University of California
Bussler, Christoph; Cisco Systems, Inc.
Charpentier, Michel; University of New Hampshire
Chong, Stephen; Cornell University
Collison, George; The Concord Consortium
DeWeaver, Eric; University of Wisconsin - Madison
Gans, Eric; University of California
Gill, Sam; San Francisco State University
Hunter, Lynette; University of California Davis
Iceland, John; University of Maryland
Kaplan, Samantha W.; University of Wisconsin
Langou, Julien; The University of Tennessee
Liu, Yuliang; Southern Illinois University Edwardsville
Lok, Benjamin; University of Florida
Minh, Chi Cao; Stanford University
Morrissey, Robert; The University of Chicago
Mui, Lik; Google, Inc
Rizzo, Albert ; University of Southern California
Rosenberg, Jonathan M. ; University of Maryland
Shaffer, Cliff ; Virginia Tech
Sherman, Elaine; Hofstra University
Snyder, David F.; Texas State University
Song, Zhe; University of Iowa
Wei, Chen; Intelligent Automation, Inc.
Yu, Zhiyi; University of California

Authors of papers are responsible for the contents and layout of their papers.

Welcome to IPSI BgD Conferences and Journals!

http://www.internetconferences.net

http://www.internetjournals.net

CIP – Katalogizacija u publikaciji
Narodna biblioteka Srbije, Beograd
ISSN 1820 – 4511
The IPSI BGD Transactions on Internet Research
COBISS.SR - ID 119128844