ΕΠΙΛΕΓΜΕΝΕΣ

ΠΤΥΧΙΑΚΕΣ & ΔΙΠΛΩΜΑΤΙΚΕΣ ΕΡΓΑΣΙΕΣ

ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Τόμος 13
ΑΘΗΝΑ 2016

ΕΠΙΛΕΓΜΕΝΕΣ

ΠΤΥΧΙΑΚΕΣ & ΔΙΠΛΩΜΑΤΙΚΕΣ ΕΡΓΑΣΙΕΣ

Τόμος 13

Εκδίδεται μία φορά το χρόνο από το:

Τμήμα Πληροφορικής και Τηλεπικοινωνιών
Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών,
Πανεπιστημιούπολη, 15784 Αθήνα

Επιμέλεια έκδοσης:

Επιτροπή Ερευνητικών και Αναπτυξιακών Δραστηριοτήτων

Θ. Θεοχάρης (υπεύθυνος έκδοσης), Καθηγητής, Τμήμα Πληροφορικής και Τηλεπικοινωνιών
Η. Μανωλάκος, Αναπληρωτής Καθηγητής, Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Γραφιστική επιμέλεια - Επιμέλεια κειμένων:

Λ. Χαλάτση, Γραφείο Προβολής, Τμήμα Πληροφορικής και Τηλεπικοινωνιών

ISSN 1792-8826

Εξώφυλλο: Έργο του καλλιτέχνη Antony Kitson για το Secret 7" Exhibition.

Φιλοτεχνήθηκε για το εξώφυλλο του δίσκου της Jessie Ware «Still Love me».

Περιεχόμενα

Πρόλογος

Ο τόμος αυτός περιλαμβάνει περιλήψεις επιλεγμένων διπλωματικών και πτυχιακών εργασιών που εκπονήθηκαν στο Τμήμα Πληροφορικής και Τηλεπικοινωνιών του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών κατά το διάστημα **01/01/2015 - 31/12/2015**. Πρόκειται για τον **13^{ο} τόμο** στη σειρά αυτή. Στόχος του θεσμού είναι η ενθάρρυνση της δημιουργικής προσπάθειας και η προβολή των πρωτότυπων εργασιών των φοιτητών του Τμήματος.

Η έκδοση αυτή είναι ψηφιακή και έχει δικό της ISSN. Αναρτάται στην επίσημη ιστοσελίδα του Τμήματος και έτσι, εκτός από τη μείωση της δαπάνης κατά την τρέχουσα περίοδο οικονομικής κρίσης, έχει και μεγαλύτερη προσβασιμότητα. Για το στόχο αυτό, σημαντική ήταν η συμβολή της Λήδας Χαλάτση που επιμελήθηκε και φέτος την ψηφιακή έκδοση και πέτυχε μια ελκυστική ποιότητα παρουσίασης, ενώ βελτίωσε και την ομοιογένεια των κειμένων.

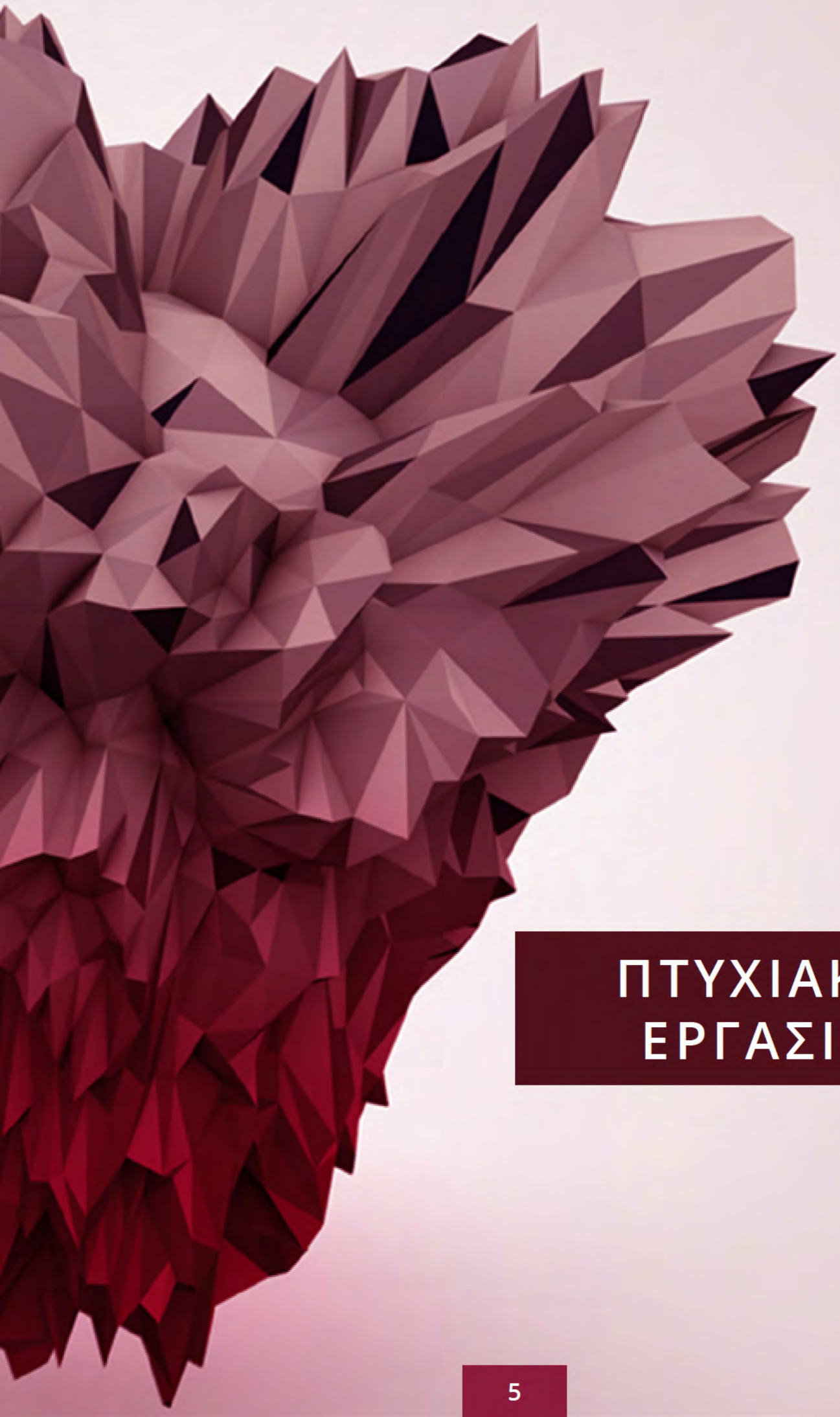
Η στάθμη των επιλεγμένων εργασιών είναι υψηλή και κάποιες από αυτές έχουν είτε δημοσιευθεί είτε υποβληθεί για δημοσίευση.

Θα θέλαμε να ευχαριστήσουμε τους φοιτητές για το χρόνο που αφιέρωσαν για να παρουσιάσουν τη δουλειά τους στα πλαίσια αυτού του θεσμού και να τους συγχαρούμε για την ποιότητα των εργασιών τους. Ελπίζουμε η διαδικασία αυτή να προσέφερε και στους ίδιους μια εμπειρία που θα τους βοηθήσει στη συνέχεια των σπουδών τους ή της επαγγελματικής τους σταδιοδρομίας.

Η Επιτροπή Ερευνητικών και Αναπτυξιακών Δραστηριοτήτων

Θ. Θεοχάρης (υπεύθυνος έκδοσης), Η. Μανωλάκος

Αθήνα, Ιούνιος 2016



ΠΤΥΧΙΑΚΕΣ ΕΡΓΑΣΙΕΣ

Improving Selfish Routing for Risk-Averse Players

Dimitrios Kalimeris (sdi1000049@di.uoa.gr)

Abstract

In this thesis we investigate how and to which extent one can exploit risk-aversion and modify the perceived cost of the players in selfish routing so that the Price of Anarchy (PoA) is improved. We adopt the model of γ-modifiable routing games, a variant of routing games with restricted tolls. We prove that computing the best γ-enforceable flow is NP-hard for parallel-link networks with affine latencies and two classes of heterogeneous risk-averse players. On the positive side, we show that for parallel-link networks with heterogeneous players and for series-parallel networks with homogeneous players, there exists a nicely structured γ-enforceable flow whose PoA improves fast as γ increases. Moreover, we prove that the PoA of this flow is best possible in the worst-case, in the sense that there are instances where (i) the best γ-enforceable flow has the same PoA, and (ii) considering more flexible modifications does not lead to any further improvement.

Keywords: Selfish Routing, Wardrop/Nash Equilibrium/ Price of Anarchy, Risk-Aversion, Heterogeneous Flows

Advisors

Vasileios Zissimopoulos, Professor, Dimitrios Fotakis, Professor NKUA and Thanasis Lianeas, PhD Candidate

1. Introduction

Routing games provide an elegant and practically useful model of selfish resource allocation in transportation and communication networks and have been extensively studied (see e.g., [11]). The majority of previous work assumes that the players select their routes based on precise knowledge of edge delays. In practical applications however, the players cannot accurately predict the actual delays due to their limited knowledge about the traffic conditions and due to unpredictable events that affect the edge delays and introduce uncertainty (see e.g., [9, 7, 1, 8] for examples). Hence, the players select their routes based only on delay estimations and are aware of the uncertainty and the potential inaccuracy of them. Therefore, to secure themselves from increased delays, whenever this may have a considerable influence, the players select their routes taking uncertainty into account (e.g., people take a safe route or plan for a longer-than-usual delay when they head to an important meeting or to catch a long-distance flight).

Recent work (see e.g., [7, 10, 1, 8] and the references therein) considers routing games with *stochastic delays* and *risk-averse players*, where instead of the route that minimizes her expected delay, each player selects a route that guarantees a reasonably low actual delay with a reasonably high confidence. There have been different models of stochastic routing games, each modeling the individual cost of risk-averse players in a slightly different way. In all cases, the actual delay is modeled as a random variable and the perceived cost of the players is either a combination of the expectation and the standard deviation (or the variance) of their delay [7, 8] or a player-specific quantile of the delay distribution [9, 1] (see also [12, 4] about the perceived cost of risk-averse players).

No matter the precise modeling, we should expect that stochastic delays and risk-aversion cannot improve the network performance at equilibrium. Interestingly, [10, 8] indicate that in certain settings, stochastic delays and risk-aversion can actually improve the network performance at equilibrium. Motivated by these results, we consider routing games on parallel-link and series-parallel networks and investigate how one can exploit risk-aversion in order to modify the perceived cost of the (possibly heterogeneous) players so that the PoA is significantly improved.

1.1. Routing Games

To discuss our approach more precisely, we introduce the basic notation and terminology about routing games. A (non-atomic) *selfish routing game* (or instance) is a tuple $\mathcal{G}=(G(V,E),(\ell_e)_{e\in E},r)$, where $G(V,E)$ is a directed network with a source s and a sink t, $\ell_e:\mathbb{R}_{\geq 0}\to\mathbb{R}_{\geq 0}$ is a non-decreasing delay (or latency function associated with edge e and $r>0$ is the traffic rate. We let $\mathcal{P}$ denote the set of simple $s-t$ paths in G. We say that G is a parallel-link network if each $s-t$ path is a single edge (or link).

A (feasible) *flow* f is a non-negative vector on $\mathcal{P}$ such that $\sum_{p\in\mathcal{P}} f_p = r$. We let $f_e=\sum_{p:e\in p} f_p$ be flow routed by f on edge e. Given a flow f, the latency of each edge e is $\ell_e(f)=\ell_e(f_e)$, the latency of each path p is $\ell_p(f)=\sum_{e\in p}\ell_e(f)$ and the latency of f is $L(f)=\max_{p:f_p>0}\ell_p(f)$.

The traffic r is divided among infinitely many players, each trying to minimize her latency. A flow f is a *Wardrop equilibrium* (or a *Nash flow*, for brevity), if all traffic is routed on minimum latency paths, i.e., for any $p\in\mathcal{P}$ with $f_p>0$ and for all $p'\in\mathcal{P}$, $\ell_p(f)\leq\ell_{p'}(f)$. Therefore, in a Nash flow f, all players incur a minimum common latency $\min_p \ell_p(f)=L(f)$. Under weak assumptions on delay functions, a Nash flow exists and is essentially unique (see e.g., [11]).

The efficiency of a flow f is measured by the *total latency* $C(f)$ of the players, i.e., by $C(f)=\sum_{e\in E} f_e\ell_e(f)$. The *optimal flow*, denoted o, minimizes the total latency among all feasible flows. The *Price of Anarchy* (PoA) quantifies the performance degradation due to selfishness. The $\mathrm{PoA}(\mathcal{G})$ of a routing game $\mathcal{G}$ is the ratio $C(f)/C(o)$, where f is the Nash flow and o is the optimal flow o of $\mathcal{G}$. The PoA of a class of routing games is the maximum PoA over all games in the class. For routing games with latency functions in a class $\mathcal{D}$, the PoA is equal to $\mathrm{PoA}(\mathcal{D})=\rho_\gamma(\mathcal{D}):=(1-\beta(\mathcal{D}))^{-1}$, where $\beta(\mathcal{D})=\sup_{\ell\in\mathcal{D},x\geq y\geq 0}\frac{y(l(x)-l(y))}{xl(x)}$ only depends on the class of latency functions $\mathcal{D}$ [11, 3].

1.2. Using Risk-Aversion to Modify Edge Latencies

The starting point of our work is that in some practical applications, we may intentionally introduce variance to edge delays so that the expected delay does not change, but the risk-averse cost of the players increases. E.g., in a transportation network, we can randomly increase or decrease the proportion of time allocated to the green traffic light for short periods or we can open or close an auxiliary traffic lane. In a communication network, we might randomly increase or decrease the link capacity allocated to a particular type of traffic or change its priority. At the intuitive level, we expect that the effect of such random changes to risk-averse players is similar to that of refundable tolls (see e.g., [5]), albeit restricted in magnitude due to the bounded variance in edge delays that we can afford.

E.g., let e be an edge with latency $\ell_e(x)$ where we can increase the latency temporarily to $(1+\alpha_1)\ell_e(x)$ and decrease it temporarily to $(1-\alpha_2)\ell_e(x)$. If we implement the former change with probability p_1 and the latter with probability $p_2 < 1-p_1$, the latency function of e becomes a random variable with expectation $\left[p_1(1+\alpha_1)+p_2(1-\alpha_2)+(1-p_1-p_2)\right]\ell_e(x)$. Adjusting p_1 and p_2 (and possibly α_1 and α_2) so that $p_1\alpha_1 = p_2\alpha_2$, we achieve an expected latency of $\ell_e(x)$. However, if the players are (homogeneously) risk-averse and their perceived delay is given by an $(1-p_1+\varepsilon)$-quantile of the delay distribution (e.g., as in [9, 1]), the perceived latency on e is $(1+\alpha_1)\ell_e(x)$. Similarly, if the individual cost of the risk-averse players are given by the expectation plus the standard deviation of the delay distribution (e.g., as in [7]), the perceived latency is $\left(1+\sqrt{p_1\alpha_1^2+p_2\alpha_2^2}\right)\ell_e(x)$. In both cases, we can achieve a significant increase in the delay perceived by risk-averse players, while the expected delay remains unchanged.

1.3. Contribution

In this work, we assume a given upper bound γ on the maximum increase in the latency functions and refer to the corresponding routing game as a *γ-modifiable game*. We consider both homogeneous and heterogeneous risk-averse players. We adopt this model as a simple and general abstraction of how one can exploit risk-aversion to improve the PoA of routing games. On the conceptual side and

to the best of our knowledge, this is the first time that risk-aversion is proposed as a means of implementing restricted tolls, and through this, as a potential remedy to the inefficiency of selfish routing.

A flow f is γ-*enforceable* if there is γ_e-modification on each edge e, with $0 \leq \gamma_e \leq \gamma$, so that f is a Nash flow of the modified game, i.e., for each player class i, for every path p used by class i, and for all paths p', $\sum_{e \in p}(1+a^i\gamma_e)\ell_e(f) \leq \sum_{e \in p'}(1+a^i\gamma_e)\ell_e(f)$. In this work, we are interested in computing either the best γ-enforceable flow, which minimizes total latency among all γ-enforceable flows, or a γ-enforceable flow with low PoA. We measure the PoA in terms of the total expected latency (instead of the total perceived delay of the players). In practical applications, the total expected latency is directly related to many crucial performance parameters (e.g., to the expected pollution in a transportation network or to the expected throughput in a communication network) and thus, it is the quantity that a central planner usually seeks to minimize.

We consider routing games on parallel links with homogeneous players and show that for every $\gamma > 0$, there is a nicely structured γ-enforceable flow whose PoA improves significantly as γ increases. gets a 0-modification, while if $f_e > o_e$, e gets a γ-modification (Lemma 1). Using the variational inequality approach of [3], we provide an asymptotically tight bound for the PoA of the network, which is a natural generalization of the bound introduced in [3].

We also investigate parallel-link games with heterogenous players. We prove that computing the best γ-enforceable flow is NP-hard for parallel-link games with affine latencies and only two classes of heterogeneous risk-averse players (Theorem 3). On the positive side, we apply [6, Algorithm 1] and show (Theorem 5) that the γ-enforceable flow f of Lemma 1 can be turned into a γ-enforceable flow for parallel-link instances with heterogeneous players. Since only the γ-modifications are adjusted for heterogeneous players, but the flow itself does not change, the PoA of f is bounded as above and remains best possible in the worst case.

Then, we extend our approach of finding a γ-enforceable flow that "mimics" the optimal flow to series-parallel networks. Extending the rerouting procedure of Lemma 1, we show that for routing games in series-parallel networks with

homogeneous players, there is a γ-enforceable flow with PoA at most $(1-\beta_{\gamma}(\mathcal{D}))^{-1}$ (Lemma 1 and Theorem 1).

Finally, we consider (p,γ)-modifiable games, where the p-norm of the edge modifications vector $(\gamma_e)_{e\in E}$ is at most γ. This generalization captures applications where the total variance introduced in the network should be bounded by γ and could potentially lead to an improved PoA. We prove that the worst-case PoA under (p,γ)-modifications is essentially identical to the worst-case PoA under $\gamma/\sqrt[p]{m}$-modifications (Theorem 7). Therefore, even for (p,γ)-modifiable games, the PoA of the $\gamma/\sqrt[p]{m}$-enforceable flow of Lemma 1 is essentially best possible. Due to space constraints, we only sketch the main ideas behind our results and defer the technical details to the full version of this work.

1.4. Previous Work

On the conceptual side, our work is closest to those considering the PoA of stochastic routing games with risk-averse players [7, 1, 10]. Nikolova and Stier-Moses [8] recently introduced the *price of risk-aversion* (PRA), which is the worst-case ratio of the total latency of the Nash flow for risk-averse players to the total latency of the Nash flow for risk-neutral players. Interestingly, PRA can be smaller than 1 and as low as $1-\beta(\mathcal{D})$ for stochastic routing games on parallel-links (i.e., risk-aversion can improve the PoA to 1 for certain instances). On the technical side, our workis closest to those investigating the properties of restricted refundable tolls for routing games [2, 6].

2. The Model and Preliminaries

The basic model of routing games is introduced in Section 1. Next, we introduce some more notation and the classes of γ-modifiable and (p,γ)-modifiable games.

2.1. γ-Modifiable Routing Games.

A selfish routing game with heterogeneous players in k classes is a tuple $\mathcal{G}=(G(V,E),(\ell_e)_{e\in E},(a^i)_{i\in[k]},(r^i)_{i\in[k]})$, where G is a directed $s-t$ network

with m edges, a^i is the aversion factor of the players in class i and r_i is the amount of traffic with aversion a^i. We assume that $a^1 = 1$ and $a^1 < a^2 < \ldots < a^k$. If the players are homogeneous, there is a single class with risk aversion $a^1 = 1$ and traffic rate r. Then, an instance is $\mathcal{G} = (G, \ell, r)$.

A flow f is a non-negative vector on $\mathcal{P} \times \{1, \ldots, k\}$. We let $f_p^{a^i}$ be the flow with aversion a^i on path p and $f_p = \sum_i f_p^{a^i}$ be the total flow on path p. Similarly, $f_e^{a^i} = \sum_{p: e \in p} f_p^{a^i}$ isthe flow with aversion a^i on edge e and $f_e = \sum_i f_p^{a^i}$ is the total flow on edge e. We let $a_e^{\min}(f)$ be the smallest aversion factor in e under f. If e is not used by f, we let $a_e^{\min}(f) = a^k$. We say that an edge e (resp. path p) is used by players of type a^i if $f_e^{a^i} > 0$ (resp. for all $e \in p$). To simplify notation, we may write ℓ_e, instead of $\ell_e(f)$.

We say that a routing game $\mathcal{G}$ is γ*-modifiable* if we can select a $\gamma_e \in [0, \gamma]$ for each edge e and change the edge latencies perceived by the players of type a^i from $\ell_e(x)$ to $(1 + a^i \gamma_e)\ell_e(x)$ using small random perturbations.

Any vector $\vec{\Gamma} = (\gamma_e)_{e \in E}$, where $\gamma_e \in [0, \gamma]$ for each edge , is a γ*-modification* of $\mathcal{G}$. Given a γ-modification $\vec{\Gamma}$, we let $\mathcal{G}^{\vec{\Gamma}}$ denote the γ-modified routing game where the perceived cost of the players is changed according to the modification $\vec{\Gamma}$.

A flow f is a *Nash flow* of $\mathcal{G}^{\vec{\Gamma}}$, if for any path p and any type a^i with $f_p^{a^i} > 0$ and for all paths p', $\sum_{e \in p}(1 + a^i \gamma_e)\ell_e(f) \leq \sum_{e \in p'}(1 + a^i \gamma_{e'})\ell_{e'}(f)$. Given a routing game $\mathcal{G}$, we say that a flow f is γ*-enforceable*, or simply *enforceable*, if there exists a γ-modification $\vec{\Gamma}$ of $\mathcal{G}$ such that f is a Nash flow of $\mathcal{G}^{\vec{\Gamma}}$.

Our assumption is that γ-modifications do not change the expected latency. Therefore, the total latency of f in both $\mathcal{G}^{\vec{\Gamma}}$ and $\mathcal{G}$ is $C(f) = \sum_{e \in E} f_e \ell_e(f)$. Hence, the optimal flow o of $\mathcal{G}$ is also an optimal flow of $\mathcal{G}^{\vec{\Gamma}}$. A flow f is the *best* γ*-enforceable* flow of $\mathcal{G}$ if for any other γ-enforceable flow f' of $\mathcal{G}$, $C(f) \leq C(f')$. The Price of Anarchy $\mathrm{PoA}(\mathcal{G}^{\vec{\Gamma}})$ of the modified game $\mathcal{G}^{\vec{\Gamma}}$ is equal to $C(f)/C(o)$, where f is the Nash flow of $\mathcal{G}^{\vec{\Gamma}}$. For a γ-modifiable game $\mathcal{G}$, the PoA of $\mathcal{G}$ under γ-modifications, denoted $\mathrm{PoA}_\gamma(\mathcal{G})$, is $C(f)/C(o)$,

where f is the best γ-enforceable flow of $\mathcal{G}$. For routing games with latency functions in class $\mathcal{D}$, $\mathrm{PoA}_\gamma(\mathcal{D})$ denotes the maximum $\mathrm{PoA}_\gamma(\mathcal{G})$ over all γ-modifiable games $\mathcal{G}$ with latencies in $\mathcal{D}$.

2.2. (p, γ)-Modifiable Routing Games

Generalizing γ-modifiable games, we select a modification $\gamma_e \geq 0$ for each edge e so that $\|(\gamma_e)_{e\in E}\|_p = \sqrt[p]{\sum_{e\in E}\gamma_e^p} \leq \gamma$, for some given integer $p \geq 1$, and change the perceived edge latencies as above. We refer to such games as (p,γ)-modifiable. All the notation above naturally generalizes to (p,γ)-modifiable games. The PoA of a game $\mathcal{G}$ under (p,γ)-modifications, denoted $\mathrm{PoA}_\gamma^p(\mathcal{G})$, is $C(f)/C(o)$, where f is the best (p,γ)-enforceable flow of $\mathcal{G}$. Similarly, $\mathrm{PoA}_\gamma^p(\mathcal{D})$ is the maximum PoA of all (p,γ)-modifiable games with latency functions in class $\mathcal{D}$.

2.3. Series-Parallel Networks

A directed s - t network $G(V,E)$ is *series-parallel* if it either consists of a single edge (s,t) or can be obtained from two series-parallel networks with terminals (s_1,t_1) and (s_2,t_2) composed either in series or in parallel. In a *series composition*, t_1 is identified with s_2, s_1 becomes s, and t_2 b comes t. In a *parallel composition*, s_1 is identified with s_2 and becomes s, and t_1 is identified with t_2 and becomes t.

3. Modifying Routing Games in Parallel-Link Networks

In this section, we study γ-modifiable games on parallel-link networks with homogeneous risk-averse players. We show that for any instance $\mathcal{G}$, there exist a flow f mimicking the optimal flow of $\mathcal{G}$, o, and a γ-modification enforcing f as the Nash flow of the modified instance.

Lemma 1. Let $\mathcal{G} = (G,\ell,r)$ be a γ-modifiable instance on parallel-links with homogeneous risk-averse players and let o be the optimal flow of $\mathcal{G}$. There is a feasible flow f and a γ-modification $\vec{\Gamma}$ of $\mathcal{G}$ such that

(i) f is a Nash flow of the modified instance $\mathcal{G}^{\vec{\Gamma}}$.

(ii) for any link e, if $f_e < o_e$, then $\gamma_e = 0$, and if $f_e > o_e$, then $\gamma_e = \gamma$.

Moreover, given o, we can compute f and $\vec{\Gamma}$ in time $O(mT_{\mathrm{NE}})$, where T_{NE} is the complexity of computing the Nash flow of any given γ-modification of $\mathcal{G}$.

Proof. The proof of the theorem is constructive, by induction on the number of links. The base case is obvious. For the inductive step, let m be a used link with maximum latency in o. Removing m and decreasing the total traffic rate by $o_m > 0$, we obtainan instance $\mathcal{G}_{-m} = (G_{-m}, \ell, r - o_m)$ with one link less than $\mathcal{G}$. By induction hypothesis, there are a flow f' and a γ-modification $\vec{\Gamma}' = (\gamma'_e)_{e \in E_{-m}}$ so that properties (i) and (ii) hold for $\mathcal{G}_{-m}$. Now we restore link m and the traffic rate to r. The lemma follows directly from the hypothesis if there is a modification γ_m so that $(1+\gamma_m)\ell_m(o) = L(f')$. Otherwise, we have that $\ell_m(o) > L(f')$. Then, we carefully reroute flow from link m to the remaining links while maintaining properties (i) and (ii) in $\mathcal{G}_{-m}$. We do so until the latency of m becomes equal to the cost of the equilibrium flow that we maintain (under rerouting) in $\mathcal{G}_{-m}$. In order to maintain property (ii), we pay attention to links e where the flow f'_e reaches for the first time and to links e' where $\gamma'_{e'}$ reaches γ for the first time. For the former, we stop increasing flow and start increasing γ'_e, so that the equilibrium property is maintained. For the latter, we stop increasing $\gamma'_{e'}$ and start increasing the flow again.

3.1. Price of Anarchy Analysis

We next prove an upper bound on the PoA of the γ-enforceable flow f of Lemma 1. This also serves as an upper bound on the PoA_γ of the best γ-enforceable flow. The approach is conceptually similar to that of [3] and exploits the properties (i) and (ii) of Lemma 1.

Theorem 1. For γ-modifiable instances on parallel-links with latency functions in class $\mathcal{D}$, $\mathrm{PoA}_\gamma(\mathcal{D}) \le \rho_\gamma(\mathcal{D}) := (1 - \beta_\gamma(\mathcal{D}))^{-1}$, where

$$\beta_\gamma(\mathcal{D}) = \sup_{\ell \in \mathcal{D}, x \ge y \ge 0} \frac{y\big(\ell(x) - \ell(y)\big) - \gamma(x-y)\ell(x)}{x\ell(x)}.$$

Furthermore, we can show in the following theorem that bounds on the PoA_γ of Theorem 1 are best possible in the worst-case.

Theorem 2. For any class of latency functions $\mathcal{D}$ and for any $\varepsilon > 0$, there is a γ-modifiable instance $\mathcal{G}$ on parallel links with homogeneous risk-averse players and latencies in class $\mathcal{D}$ so that $\mathrm{PoA}_\gamma(\mathcal{G}) \geq \rho_\gamma(\mathcal{D}) - \varepsilon$.

4. Parallel-Link Games with Heterogeneous Players

In contrast to the case of homogeneous players, computing the best γ-enforceable flow for heterogeneous risk-averse players is NP-hard, even for affine latencies.

Theorem 3. Given an instance $\mathcal{G}$ on parallel links with affine latencies and two classes of risk-averse players, a $\gamma > 0$ and a target cost $C > 0$, it is NP-complete to determine whether there is a γ-enforceable flow with total latency at most C.

4.1. γ-Enforceable Flows with Good Price of Anarchy

Since the best enforceable flow is NP-hard, we next establish the existence of an enforceable flow that "mimics" the optimal flow o, as described by the properties (i) and (ii) in Lemma 1 and achieves a PoA as low as that in Theorem 1. In the following, we assume that the links are indexed in increasing order of $\ell_i(f)$, i.e. $i < j \Rightarrow \ell_i(f) \leq \ell_j(f)$, with ties broken in favor of links with $f_e > 0$. We start with a necessary and sufficient condition for a flow f to be γ-enforceable. [6, Algorithm 1] shows how to efficiently compute a γ-modification for any flow f that satisfies the following.

Theorem 4. ([6, Theorem 5]) Let $\mathcal{G}$ be a γ-modifiable instance on parallel links with heterogeneous players, let f be a feasible flow and let μ be the maximum index of a link used by f. Then, f is γ-enforceable if and only if

(i) for any used link i, $\gamma \ell_i(f) \geq \sum_{l=i}^{\mu-1} \frac{\ell_{l+1}(f) - \ell_l(f)}{a_{l+1}^{min}}$ and

(ii) for all links i and j, if $\ell_i(f) < \ell_j(f)$, then $a_i^{\max}(f) \leq a_j^{\min}(f)$ (more risk-averse players use links of higher latency).

To obtain a γ-enforceable flow f for an instance with heterogeneous players, we combine Lemma 1 with Theorem 4 and apply [6, Algorithm 1]. Specifically, we first ignore player heterogeneity and compute, using Lemma 1, a γ-enforceable flow and the corresponding modification $\vec{\Gamma}$ so that f is a Nash flow of the modified game $\mathcal{G}^{\vec{\Gamma}}$ when all players have the minimum risk-aversion factor $a^1 = 1$. Assuming that the links are indexed in increasing order of their latencies in f, since f is γ-enforceable with risk-aversion factor $a^1 = 1$ for all players, Theorem 4 implies that for any used link i, $(1+\gamma)\ell_i(f) \geq \ell_\mu(f)$.

Next, we greedily allocate the heterogeneous risk-averse players to f, taking their risk-averse factors into account, so that each link i receives flow f_i and property (ii) in Theorem 4 is satisfied. Finally, we use [6, Algorithm 1] and compute a γ-modification that turns f into an equilibrium flow for the modified instance with heterogeneous players. This is possible because, by construction, f satisfies condition (i) of Theorem 4. Moreover, since f satisfies the properties of (i) and (ii) in Lemma 1, the PoA of f can be bounded as in Theorem 1. Hence, we obtain the following.

Theorem 5. Let $\mathcal{G}$ be a γ-modifiable instance on parallel-links with heterogeneous risk-averse players. Given the optimal flow of $\mathcal{G}$, we can compute a feasible flow f and a γ-modification $\vec{\Gamma}$ of $\mathcal{G}$ in time $O(mT_{\mathrm{NE}})$, where T_{NE} is the complexity of computing the Nash flow of any given γ-modification of $\mathcal{G}$ with homogeneous risk-averse players. Moreover, the PoA_γ, under γ-modifications, achieved by f is upper bounded as in Theorem 1.

5. Modifying Routing Games in Series-Parallel Networks

In this section, we consider γ-modifiable instances on series-parallel networks with homogeneous players and generalize the results of Section 3. We proceed

to generalized Lemma 1 to series-parallel networks. The proof is based on an extension of the rerouting procedure in Lemma 1 combined with a continuity property of γ-enforceable flows in series-parallel networks.

Lemma 2. Let $\mathcal{G}=(G,\ell,r)$ be a γ-modifiable instance with homogeneous risk-averse players on a series-parallel network G and let o be the optimal flow of $\mathcal{G}$. There is a feasible flow f and a γ-modification $\vec{\Gamma}$ of $\mathcal{G}$ such that

(i) f is a Nash flow of the modified instance $\mathcal{G}^{\vec{\Gamma}}$.

(ii) for any edge e, if $f_e < o_e$, then $\gamma_e = 0$, and if $f_e > o_e$, then $\gamma_e = \gamma$.

Using the properties (i) and (ii), we show that the upper bound on the PoA in Theorem 1 extends to the γ-enforceable flow f of Lemma 2 and to the PoA_γ of the best γ-enforceable flow in series-parallel networks with homogeneous players.

Theorem 6. For γ-modifiable instances on series-parallel networks with homogeneous players and latency functions in class $\mathcal{D}$, $\mathrm{PoA}_\gamma(\mathcal{D}) \le \rho_\gamma(\mathcal{D})$.

Given the optimal flow of an instance $\mathcal{G}$ on a series-parallel network, we show how to compute a γ-enforceable flow f and the corresponding modification so that we achieve a PoA at most $\rho_\gamma(\mathcal{D})$. Given o, the running time is determined by the time required to compute a Nash flow of the original instance.

We first determine whether the optimal flow o is γ-enforceable. To this end, we remove from G all edges unused by o and check the feasibility of the following:

$$\begin{aligned} 0 \le \gamma_e \le \gamma, \qquad & \forall \text{ used edges } e \\ \sum_{e\in p}(1+\gamma_e)\ell_e(o) = \max_{p:o_p>0}\ell_p(o) \qquad & \forall \text{ used path } p \end{aligned} \tag{1}$$

If the linear system (1) is not feasible, then o is not γ-enforceable. Otherwise, using the solution of (1) as γ_e's for the edges of G used by o and setting $\gamma_e = 0$ for the unused edges e, we enforce o as a Nash flow of the modified game $\mathcal{G}^{\vec{\Gamma}}$.

If (1) is not feasible and o is not γ-enforceable, we exploit the constructive

nature of the proof of Lemma 2 and find a γ-enforceable flow in time dominated by the time required to compute a Nash flow in series-parallel networks.

Lemma 3. Let $\mathcal{G}$ be a γ-modifiable instance on a series-parallel network with homogeneous players. Given the optimal flow of $\mathcal{G}$ and any $\varepsilon > 0$, we can compute a feasible flow f and a γ-modification $\mathcal{G}^{\vec{\Gamma}}$ of $\mathcal{G}$ with the properties (i) and (ii) of Lemma 2 in time $O\left(m^2 T_{\mathrm{NE}} \log\left(r/\varepsilon\right)\right)$, where T_{NE} is the complexity of computing the Nash flow of any given γ-modification of $\mathcal{G}$ and ε is an accuracy parameter.

6. Parallel-Link Games with Relaxed Restrictions

In this section, we consider (p,γ)-modifiable games on parallel links with heterogeneous risk-averse players. Observing that any $\gamma / \sqrt[p]{m}$-modification is a (p,γ)-modification for a (p,γ)-modifiable game, we next show an upper bound on the PoA under such modifications.

Theorem 7. For any (p,γ)-modifiable instance $\mathcal{G}$ on m parallel links with heterogeneous risk-averse players and latency functions in class $\mathcal{D}$, we have that $\mathrm{PoA}_{\gamma}^{p}(\mathcal{G}) \le \mathrm{PoA}_{\gamma_0}(\mathcal{G}) \le \rho_{\gamma_0}(\mathcal{D})$, where $\gamma_0 = \gamma / \sqrt[p]{m}$.

The above bound is tight under weak assumptions on the class $\mathcal{D}$ of latency functions. More specifically, we say that a class of latency functions $\mathcal{D}$ is of the form $\mathcal{D}_0$ if (a) ℓ is continuous and twice differentiable in $(0,+\infty)$, (b) $\ell'(x) > 0, \forall x \in (0,+\infty)$ or ℓ is constant, (c) ℓ is semi-convex, i.e. $x\ell(x)$ is convex in $[0,+\infty)$ and (d) if $\ell \in \mathcal{D}$, then $(\ell + c) \in \mathcal{D}$, for all constants $c \in \mathbb{R}$ such that for all $x \in \mathbb{R}_{\ge 0}$, $\ell(x) + c \ge 0$.

Then we obtain the following.

Theorem 8. For any class $\mathcal{D}$ of the form $\mathcal{D}_0$ and any $\varepsilon > 0$ there is an instance $\mathcal{G}$ on m parallel links with homogeneous players and latency functions in class $\mathcal{D}$, so that $\mathrm{PoA}_{\gamma}^{p}(\mathcal{G}) \ge \rho_{\gamma_0}(\mathcal{D}) - \varepsilon$, where $\gamma_0 = \gamma / \sqrt[p]{m}$.

Proof. We consider an instance $\mathcal{I}_m$, with m parallel links, where the first $m-1$ links have the same latency function $\ell \in \mathcal{D}$ (to be fixed later) and link m has constant latency $(1+\gamma_1)\ell\left(\frac{r}{m-1}\right)$, where $\gamma_1 = \gamma / \sqrt[p]{m-1}$. The instance has homogeneous risk-averse players with risk-aversion $a^1 = 1$. Also we let $\gamma_0 = \gamma / \sqrt[p]{m}$. The proof is an immediate consequence of the following three claims:

Claim 1. For every $m \geq 2$ and any latency function $\ell \in \mathcal{D}$ with $\ell(0) = 0$, $\mathrm{PoA}_{\gamma}^{p}(\mathcal{I}_m) = \mathrm{PoA}_{\gamma_1}(\mathcal{I}_m)$. I.e., Claim 1 states that the best (p,γ)-modification for the instance $\mathcal{I}_m$ is the modification that splits γ evenly among the first $m-1$ edges. The proof follows from an application of KKT optimality conditions.

Claim 2. For every $m \geq 2$ and any $\varepsilon > 0$, there is a latency function $\ell_{\varepsilon,m}$ with $\ell_{\varepsilon,m}(0) = 0$ such that setting $\ell = \ell_{\varepsilon,m}$ in the instance $\mathcal{I}_m$ results in $\mathrm{PoA}_{\gamma_1}(\mathcal{I}_m) \geq (1-\beta_{\gamma_1}(\mathcal{D}))^{-1} - \varepsilon/2$. The proof of Claim 2 is similar to the proof of Theorem 2.

Since $\ell_{\varepsilon,m}(0) = 0$, we can combine claims 1 and 2 and obtain that for any $m \geq 2$ and any $\varepsilon > 0$, $\mathrm{PoA}_{\gamma}^{p}(\mathcal{I}_m) \geq \rho_{\gamma_1}(\mathcal{D}) - \varepsilon/2$, if we use the latency function $\ell_{\varepsilon,m}$.

Claim 3. For every class of latency functions $\mathcal{D}$, any $\varepsilon > 0$ and any γ, there exists an $m_\varepsilon \geq 2$ such that $\rho_{\gamma_1}(\mathcal{D}) \geq \rho_{\gamma_0}(\mathcal{D}) - \varepsilon/2$.

The proof is based on the fact that γ_1 tends to γ_0 as the number of parallel links m grows. Therefore, for any $\varepsilon > 0$, there are an m_ε and a latency function $\ell_{\varepsilon,m_\varepsilon}$ such that $\mathrm{PoA}_{\gamma}^{p}(\mathcal{I}_{m_0}) \geq \rho_{\gamma_0}(\mathcal{D}) - \varepsilon$.

7. Conclusions and Future Work

Although the model we proposed is simple and general, our results mainly apply on specially structured networks. The main question to answer is whether we can we leverage the uncertainty in more general networks and draw meaning-

ful results on improving the price of anarchy.

Especially for the case of heterogeneous players one can wonder if this connection between the performance guarantees of the homogeneous and heterogeneous flows remains the same when we face more complicated networks than parallel arcs. We have (almost) proved that for extension-parallel networks the performance guarantee we can attain is the same for both homogeneous and heterogeneous players but even for the case of series-parallel networks the answer is vague.

Another very interesting question is whether we can exploit the uncertainty to improve network's performance in cases where the path costs are not additive, like in the model presented in [7].

References

[1] H. Angelidakis, D. Fotakis, and T. Lianeas. Stochastic congestion games with risk-averse players. In *Proc. of the 6th Symposium on Algorithmic Game Theory (SAGT '13)*, LNCS 8146, pp. 86-97, 2013.

[2] V. Bonifaci, M. Salek, and G. Schäfer. Efficiency of restricted tolls in non-atomic network routing games. In *Proc. of the 4th Symposium on Algorithmic Game Theory (SAGT '10)*, LNCS 6982, pp. 302-313, 2011.

[3] J.R. Correa, A.S. Schulz, and N.E. Stier Moses. Selfish Routing in Capacitated Networks. *Mathematics of Operations Research*, 29(4):961-976, 2004.

[4] A. Fiat and C.H. Papadimitriou. When the players are not expectation maximizers. In *Proc. of the 3th Symposium on Algorithmic Game Theory (SAGT '10)*, LNCS 6386, pp. 1-14, 2010.

[5] L. Fleischer, K. Jain, and M. Mahdian. Tolls for Heterogeneous Selfish Users in Multicommodity Networks and Generalized Congestion Games. In *Proc. of the 45th IEEE Symp. on Foundations of Computer Science (FOCS '04)*, pp. 277-285, 2004.

[6] T. Jelinek, M. Klaas, and G. Schäfer. Computing optimal tolls with arc restrictions and heterogeneous players. In *Proc. of the 31st Symposium on Theoretical Aspects of Computer Science (STACS '14)*, LIPIcs 25, pp. 433-444, 2014.

[7] E. Nikolova and N. Stier Moses. Stochastic selfish routing. In *Proc. of the 4th*

Symposium on Algorithmic Game Theory (SAGT '11), LNCS 6982, pp. 314-325, 2011.

[8] E. Nikolova and N. Stier-Moses. The burden of risk aversion in mean-risk selfish routing. In *Proc. of the 16th ACM Conference on Electronic Commerce (EC '15)*, pp. 489-506, 2015.

[9] F. Ordóñez and N. Stier Moses. Wardrop equilibria with risk-averse users. *Transportation Science*, 44(1):63-86, 2010.

[10] G. Piliouras, E. Nikolova, and J.S. Shamma. Risk Sensitivity of Price of Anarchy under Uncertainty. In *Proc. of the 14th ACM Conference on Electronic Commerce (EC '13)*, pp. 715-732, 2013.

[11] Tim Roughgarden. *Selfish routing and the Price of Anarchy*. MIT press, 2005.

[12] A. Tversky and D. Kahneman. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263-291, 1979.

Διαμοιρασμός Πόρων ως Υπηρεσία : Τεχνικές Προκλήσεις και Λύσεις για Ασύρματα Δίκτυα Πέμπτης Γενιάς

Μαρία-Ευγενία Ι. Ξεζωνάκη (mxezonaki@di.uoa.gr)

Περίληψη

Η παρούσα εργασία έχει ως στόχο την εκτενή επισκόπηση, μελέτη και ανάλυση των συστημάτων κινητών επικοινωνιών 5ης γενιάς (5G) που αναμένεται να αναπτυχθούν στα επόμενα έτη καθώς και την ανάπτυξη λύσης για το διαμοιρασμό ραδιοπόρων (Radio Resource Sharing - RRS) σε δίκτυα 5G. Μετά την προσεκτική μελέτη ενός προτεινόμενου πρωτοκόλλου διαμοιρασμού πόρων το οποίο θέτει στο επίκεντρο την πλευρά του δικτύου, προτείνεται ένα νέο μοντέλο διαφορετικής προσέγγισης, θέτοντας στο επίκεντρο την πλευρά της κινητής συσκευής. Το εν λόγω μοντέλο δύναται να επιτύχει την ανάθεση των περισσότερων ευθυνών στα κινητά και την ελαχιστοποίηση της προσαρμογής του δικτύου στις ανάγκες των κινητών συσκευών.

Λέξεις κλειδιά: Συστήματα επικοινωνιών πέμπτης γενιάς, Διαμοιρασμός ραδιο-πόρων, Δικτύωση βασισμένη στο λογισμικό, Προσέγγιση από την πλευρά του δικτύου, Προσέγγιση από την πλευρά της συσκευής.

Επιβλέπων

Λάζαρος Μεράκος, Καθηγητής

1. Εισαγωγή

Τα τελευταία χρόνια έχει σημειωθεί ραγδαία άνοδος του κλάδου των κινητών επικοινωνιών, αφού η χρήση των κινητών συσκευών εξαπλώνεται με ταχύτατους ρυθμούς και αναμένεται να συνεχίσει τη διείσδυσή της στην καθημερινότητα των καταναλωτών. Η ολοένα αυξανόμενη ζήτηση για νέες υπηρεσίες και υψηλότερους ρυθμούς μετάδοσης, το cloud computing και η σύνδεση πολλών και διαφορετικών τύπων συσκευών, ωθούν τα σημερινά δίκτυα κινητών επικοινωνιών στα όρια των αντοχών τους.

Το γεγονός αυτό καθιστά αναγκαία την ανάπτυξη νέων δικτύων με αυξημένες δυνατότητες, ώστε να είναι δυνατή η εξυπηρέτηση των χρηστών με την καλύτερη δυνατή ποιότητα υπηρεσίας, τη μικρότερη δυνατή καθυστέρηση και ταυτόχρονα τη βέλτιστη αξιοποίηση των πόρων του δικτύου. Παράλληλα, ενώ η ανάγκη για απόκτηση νέων πόρων αυξάνεται, αλλά φαντάζει αδύνατη και ζημιογόνα, παρατηρούνται φαινόμενα κατασπατάλησης των ήδη υπαρχόντων πόρων. Η κατανομή και χρήση των δικτυακών πόρων με αποδοτικό τρόπο κρίνεται ζήτημα υψηλής σημασίας και στόχος της εργασίας είναι η μελέτη προσεγγίσεων για την διευθέτησή του καθώς και η πρόταση ενός νέου πρωτοκόλλου.

Σχετικές μελέτες έχουν διεξαχθεί από τον M. Yang και την ομάδα του [1], όπου εισάγεται μία OpenRAN αρχιτεκτονική η οποία βασίζεται στην τεχνολογία SDN και εκμεταλλεύεται τα πλεονεκτήματα των ετερογενών δικτύων. Επίσης, η χρήση της εν λόγω τεχνολογίας ως μέσο για την απλοποίηση της διαχείρισης των κυψελωτών δικτύων αναλύεται στην έρευνα του Κ. Πεντικούση και των συνεργατών του [2]. Μία απλή λύση για το διαμοιρασμό πόρων έχει προταθεί από την ερευνητική ομάδα του J. Panchal [3]. Η πρόταση αυτή αφορά στην παροχή on demand υποδομής και διαμοιρασμού φάσματος μεταξύ διαφορετικών παρόχων. Ακόμη, συγκεκριμένη αρχιτεκτονική δικτύου αλλά και συγκεκριμένο πρωτόκολλο σηματοδοσίας για την υλοποίηση του διαμοιρασμού των ραδιοπόρων έχουν προταθεί από τον Δ. Ξενάκη και τους συνεργάτες του [4]. Η συγκεκριμένη προσέγγιση εστιάζει περισσότερο στην πλευρά στου δικτύου (network-centric approach), δηλαδή το δίκτυο είναι η οντότητα η οποία αναλαμβάνει την πλειοψηφία των ευθυνών για την υλοποίηση της υπηρεσίας RRS. Η τελευταία εργασία μελετήθηκε διεξοδικά και με βάση αυτή, στην παρούσα εργασία προτείνεται ένα πρωτόκολλο εναλλακτικής προσέγγισης και διεξάγεται συγκριτική ανάλυση μεταξύ των δύο προσεγγίσεων.

2. Προσέγγιση από την Πλευρά του Δικτύου (Network-Centric Approach)

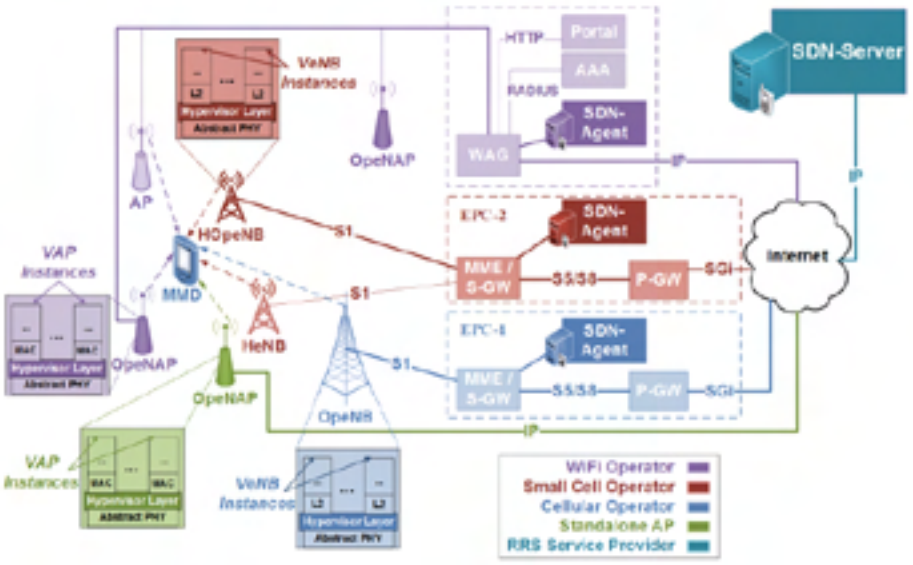
Ένα σύστημα κινητών επικοινωνιών 5ης γενιάς αναμένεται να αποτελεί συνδυασμό από αρχιτεκτονικές συστημάτων παλαιότερων γενεών και όχι μία εντελώς νέα ανεξάρτητη αρχιτεκτονική, τουλάχιστον ως προς το φυσικό επίπεδο. Άλλωστε, ακόμη και τα σημερινά συστήματα κινητών επικοινωνιών είναι ετερογενή, αποτελούνται δηλαδή από συνδυασμό δικτύων διαφορετικών αρχιτεκτονικών. Ένα αντιπροσωπευτικό παράδειγμα αρχιτεκτονικής 5G αποτελείται από ένα σύστημα μακροκυψελών LTE-A, ένα σύστημα μικρών κυψελών (picocells ή femptocells) LTE-A, ένα Wi-Fi σύστημα και ένα απομονωμένο Access Point (AP). Η εργασία [4] τονίζει ότι απαιτούνται κάποια ακόμη νέα αρχιτεκτονικά στοιχεία και λειτουργικότητες που ξεφεύγουν από τη δομή της αρχιτεκτονικής των συστημάτων LTE-A και Wi-Fi έτσι ώστε να υποστηριχθεί η προτεινόμενη ιδέα του διαμοιρασμού των ραδιο-πόρων για τα συστήματα 5ης γενιάς.

Η πρώτη αρχιτεκτονική εξέλιξη που εισάγεται είναι ο εξοπλισμός κάποιων από τους (H)eNBs και APs (με HeNBs να συμβολίζει τους eNBs του οικιακού δικτύου κάθε συνδρομητή) με δυνατότητες τεχνολογίας SDN, ώστε να μπορούν να κατανέμουν το πλεόνασμα των ραδιοπόρων τους σε χρήστες οι οποίοι είναι εγγεγραμμένοι σε διαφορετικό δίκτυο. Αυτός ο τύπος (H)eNBs και APs αναφέρονται ως (H)OpeNBs και OpenAPs αντίστοιχα. Για να επιτευχθεί κάτι τέτοιο εισάγεται ένα επιπρόσθετο επίπεδο στη στοίβα πρωτοκόλλων, το οποίο καλείται Hypervisor επίπεδο. Το εν λόγω επίπεδο παρεμβάλλεται ανάμεσα στο φυσικό επίπεδο και στα ανώτερά του και εξυπηρετεί τους (H)eNBs ώστε να εικονικοποιήσουν τους φυσικούς ραδιοπόρους τους. Προκειμένου να μπορεί να λειτουργήσει σωστά το επίπεδο του Hypervisor, εισάγεται μία επιπλέον δυνατότητα στους (H)OpeNBs. Πρόκειται για τη δυνατότητα να δημιουργούν πολλαπλά στιγμιότυπα λογισμικού από το δεύτερο επίπεδο της στοίβας πρωτοκόλλων τους και πάνω. Κάθε τέτοιο στιγμιότυπο ονομάζεται εικονικό eNB (virtual eNB - VeNB), και έχει τη δυνατότητα να προσομοιώνει τη λειτουργικότητα ενός (H)eNB ο οποίος ανήκει σε διαφορετικό δίκτυο, καθώς και να διασυνδέεται στο δίκτυο κορμού του οικιακού δικτύου μέσω ενός tunnel του τρίτου επιπέδου της στοίβας πρωτοκόλλων.

Μία ακόμη αρχιτεκτονική και λειτουργική καινοτομία που προτείνεται να εισα-

χθεί προκειμένου να υλοποιηθεί η δυνατότητα RRS είναι η εγκατάσταση μίας οντότητας που θα διαθέτει το ρόλο ελεγκτή SDN και θα ονομάζεται πράκτορας SDN (SDN Agent) στο δίκτυο κάθε παρόχου. Ένας SDN Agent είναι υπεύθυνος για την κεντρική διαχείριση της δημιουργίας VeNBs καθώς και για την προσαρμογή των πολιτικών RRS στο επίπεδο Hypervisor του παρόχου στον οποίο ανήκει ανάλογα με τα αιτήματα σχετικών με RRS των άλλων παρόχων. Επίσης, ο SDN Agent είναι επιφορτισμένος με τη διατήρηση μίας σφαιρικής εικόνας της κατάστασης του δικτύου πρόσβασης.

Το τελευταίο στοιχείο που αναφέρεται ότι είναι απαραίτητο να προστεθεί στην εν λόγω αρχιτεκτονική είναι η οντότητα που καλείται SDN Server. Παρά το γεγονός ότι κάθε SDN Agent είναι ελεγκτής στο δίκτυο όπου είναι εγκατεστημένος, ο SDN Server είναι υπεύθυνος για την υλοποίηση της δυνατότητας RRS ανάμεσα στους διαφορετικούς τύπους δικτύων από τα οποία αποτελείται το 5G δίκτυο. Ο SDN Server συλλέγει πληροφορίες σχετικά με τη διαθεσιμότητα πόρων ανά πάροχο δικτύου, παρακολουθεί την κατάσταση των διάφορων λειτουργιών των δικτύων, εξυπηρετεί την εκτέλεση όλων των απαραίτητων λειτουργιών για τις χρεώσεις των συνδρομητών και ελέγχει τις επιπτώσεις που ενδέχεται να προκύψουν από τις υπογεγραμμένες συμφωνίες για παροχή υπηρεσιών μεταξύ των παρόχων (SLAs). Τέλος, ο SDN Server μπορεί επιπλέον να παίξει το ρόλο του ρυθμιστή για τις χρεώσεις των συνδρομητών, επιτρέποντας στους δικτυακούς παρόχους να συναγωνιστούν σε πραγματικό χρόνο για την κατανομή του φάσματος. Παρακάτω αναπαρίσταται σχηματικά η μορφή της network-centric αρχιτεκτονικής.



Εικόνα 1: Network-centric αρχιτεκτονική

Θεωρώντας την παραπάνω αρχιτεκτονική, η ερευνητική εργασία [4] προτείνει ένα πρωτόκολλο RRS θέτοντας ως επίκεντρο το ίδιο το δίκτυο (network-centric approach) και προσδιορίζεται επακριβώς η ροή της σηματοδοσίας που απαιτείται για την υλοποίησή του. Η εν λόγω σηματοδοσία ξεκινά να πραγματοποιείται για πρώτη φορά όταν μία κινητή συσκευή που είναι εγγεγραμμένη σε ένα δικτυακό πάροχο Α και εξυπηρετείται από έναν eNB που ανήκει σε αυτόν μεταφέρεται σε έναν OpeNB που ανήκει σε έναν άλλο δικτυακό πάροχο Β, με άλλα λόγια όταν πραγματοποιείται μία μεταπομπή (Handover - HO).

Ως πρώτο βήμα της διαδικασίας σηματοδοσίας μέσα στο δίκτυο, ο eNB του οικιακού δικτύο που εξυπηρετεί την κινητή συσκευή ζητά από αυτή να πραγματοποιήσει ένα σύνολο από προσδιορισμένες φυσικές μετρήσεις, στέλνοντάς της ένα μήνυμα διαμόρφωσης μετρήσεων «Measurement Configuration». Ως απάντηση στο μήνυμα αυτό και ως δεύτερο βήμα της σηματοδοσίας, η κινητή συσκευή αποστέλλει στον eNB τα αποτελέσματα των εν λόγω μετρήσεων και σύμφωνα με αυτά ο eNB αποφασίζει εάν απαιτείται HO σε κάποιον OpeNB. Τέτοιου τύπου HOs είναι βασισμένοι στην εικονικοποίηση δικτύου, εφόσον ενδέχεται να πραγματοποιηθεί μεταφορά κινητής συσκευής ακόμα και σε σταθμό βάσης άλλου δικτυακού παρόχου, συνεπώς αναφέρονται ως NV-HOs. Αν τα κριτήρια που έχουν τεθεί για ενεργοποίηση μίας NV-HO ικανοποιούνται, τότε ο οικιακός eNB επικοινωνεί με τον SDN Agent Home προωθώντας του όλες τις απαραίτητες πληροφορίες σχετικά με την κινητή συσκευή, συμπεριλαμβανομένων των μετρήσεων που αυτή πραγματοποίησε κατά την έναρξη της διαδικασίας. Με τη σειρά του ο SDN Agent, διαθέτοντας σφαιρική άποψη σχετικά με την κατάσταση του δικτύου, προσδιορίζει τον OpeNB του ξένου παρόχου (πάροχος Β) ο οποίος ικανοποιεί ένα σύνολο από προκαθορισμένα κριτήρια βασισμένα στην εικονικοποίηση του δικτύου για τη λήψη απόφασης.

Αφού ληφθεί η προαναφερθείσα απόφαση, ο SDN Agent Home στέλνει στον SDN Agent Host (δηλαδή στον SDN Agent του ξένου παρόχου) ένα αίτημα για πρόσβαση στον εν λόγω OpeNB, το οποίο ονομάζεται OpeNB Access Request και περιλαμβάνει την ταυτότητα του οικιακού παρόχου (δηλαδή του παρόχου Α), την ταυτότητα του OpeNB που προσδιορίστηκε ως κατάλληλος καθώς και οποιαδήποτε πληροφορία σχετίζεται με τα χρακτηριστικά των συνδέσεων της κινητής συσκευής οι οποίες βρίσκονται σε εξέλιξη. Ο ξένος πάροχος Β πιστοποιεί ότι το εισερχόμενο αίτημα για NV-HO συνάδει με τις συμφωνίες SLA που έχουν συναφθεί μεταξύ των δύο παρόχων και στη συνέχεια ο SDN Agent Home

προωθεί το OpeNB Access Request στον OpeNB που προσδιορίστηκε, ο οποίος αποφαίνεται σχετικά με τη δυνατότητά του ή όχι να εξυπηρετήσει μία νέα σύνδεση.

Στην περίπτωση επιτυχούς αποδοχής της σύνδεσης με την κινητή συσκευή, ο OpeNB ειδοποιεί τον SDN Agent Host ότι διαθέτει τη δυνατότητα να εξυπηρετήσει τις συνδέσεις της κινητής συσκευής μέσω ενός μηνύματος επιβεβαίωσης «OpeNB Access ACK, ο οποίος με τη σειρά του απαντάει στον OpeNB με ενα μήνυμα «VeNB instance request» για δημιουργία ενός εικονικού στιγμιοτύπου VeNB το οποίο θα αφιερώθει στον οικιακό πάροχο. Όταν ο εν λόγω OpeNB δεσμεύσει το σύνολο των απαιτούμενων πόρων και δημιουργήσει τοπικά το στιγμιότυπο VeNB για τον οικιακό πάροχο, επιβεβαιώνει και τις δύο του αυτές ενέργειες στον SDN Agent Host. Στη συνέχεια, ο SDN Agent Host αιτείται για την εγκαθίδρυση ενός tunnel επιπέδου δικτύου (L3 tunnel) μεταξύ του στιγμιοτύπου VeNB που έχει δημιουργηθεί στον OpeNB και του δικτύου κορμού του οικιακού παρόχου, στον οποίο ταυτόχρονα επιβεβαιώνει και τη δέσμευση των πόρων που απαιτούνταν. Το επόμενο βήμα πραγματοποιείται από τον SDN Agent Home, ο οποίος αποστέλλει στις οντότητες MME/S-GW του οικιακού παρόχου ένα αίτημα «L3 tunnel configuration request» για τη διαμόρφωση του εν λόγω tunnel, το οποίο και εγκαθιδρύεται μεταξύ των MME/S-GW του οικιακού δικτύου και του VeNB που δημιουργήθηκε στον OpeNB.

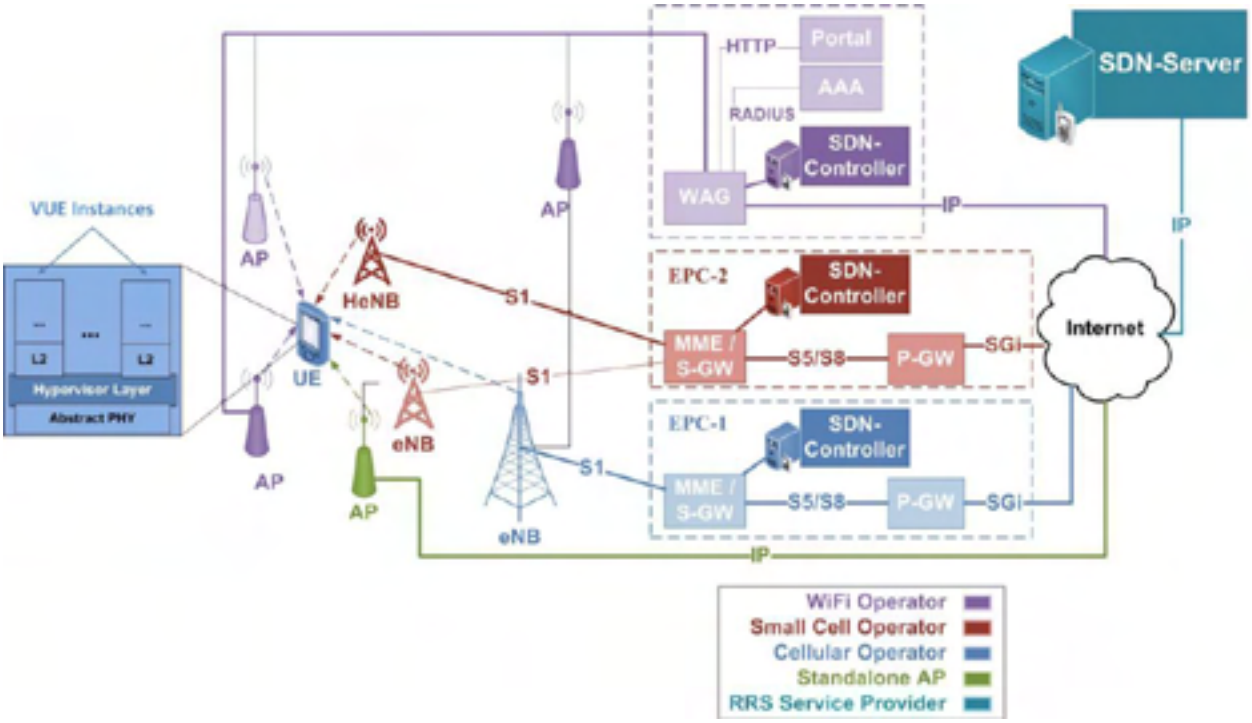
Ύστερα από την εγκαθίδρυση του L3 tunnel, ο VeNB προωθεί ένα μήνυμα «NV-HO ACK» στον eNB του οικιακού παρόχου μέσω του εν λόγω tunnel, το οποίο όταν παραλάβει ο eNB αποστέλλει εντολή μεταπομπής (HO command) στην κινητή συσκευή. Σημειώνεται ότι έως τη συγκεκριμένη στιγμή, η κινητή συσκευή δεν είχε λάβει κανενός είδους ενημέρωση για τη σηματοδοσία που εκτελούνταν και για τη διεκπεραίωση των απαιτούμενων ενεργειών για τη μεταπομπή. Τέλος, η κινητή συσκευή εκτελεί τις κλασικές διαδικασίες εκτέλεσης μίας μεταπομπής προς τον OpeNB, αφού θεωρεί τον VeNB που έχει εγκατασταθεί στον OpeNB σαν κάποιον eNB ο οποίος ανήκει στο δίκτυο του οικιακού του παρόχου (δηλαδή του παρόχου Α).

3. Προσέγγιση από την Πλευρά της Συσκευής (Device-Centric Approach) και Πρόταση Πρωτοκόλλου RRS

Η network-centric προσέγγιση RRS, παρά την αποτελεσματικότητα και τα πολλαπλά πλεονεκτήματα της, επιβαρύνει σημαντικά τα δίκτυα των παρόχων με επιπρόσθετη σηματοδοσία, καθώς οι όλες οι αποφάσεις σχετικά με την εγκαθίδρυση και αποδέσμευση των εικονικών σταθμών βάσης, και τελικά η ίδια η εγκαθίδρυση και αποδέσμευση, λαμβάνουν χώρα στα δίκτυα των συνεργαζόμενων παρόχων. Κάτι τέτοιο, δίνει με τη σειρά του το έναυσμα για εξεύρεση μίας λύσης με το ίδιο τεχνολογικό υπόβαθρο, η οποία, ωστόσο, θα φροντίζει για τη μείωση της μεταδιδόμενης σηματοδοσίας. Στην κατεύθυνση αυτή η προσοχή στρέφεται στις κινητές συσκευές. Με βάση τη νέα αυτή προσέγγιση, το σημείο απόφασης μετατοπίζεται από το σταθμό βάσης στην κινητή συσκευή, καθώς η συσκευή θα είναι αυτή που με κριτήριο τις μετρήσεις της, αλλά και πληροφορίες που τις προσφέρονται από το δίκτυο, θα λαμβάνει την απόφαση για την εξυπηρέτηση της από το σταθμό βάσης, κάποιου άλλου παρόχου.

Η προτεινόμενη δομή που θεωρείται στην παρούσα εργασία ότι πρέπει να ακολουθήσει η αρχιτεκτονική του δικτύου για την υλοποίηση του device-centric μοντέλου διαφέρει σε αρκετά σημεία από την αντίστοιχη του network-centric. Σε αντιδιαστολή με τη network-centric, στη device-centric προσέγγιση οι σταθμοί βάσης των διάφορων τύπων δικτύων (συστήματα μακροκυψελών LTE-A, συστήματα μικρών κυψελών LTE-A, συστήματα Wi-Fi και μεμονωμένα AP) δεν εξοπλίζονται με δυνατότητες τεχνολογίας SDN, ούτε και εισάγεται επίπεδο Hypervisor στη στοίβα πρωτοκόλλων των σταθμών βάσης. Όμως, είναι απαραίτητος ο εξοπλισμός των κινητών συσκευών με δυνατότητες τεχνολογίας SDN, έτσι ώστε μέσω κατάλληλης προσαρμογής να μπορούν να συνδέονται σε διαφορετικούς παρόχους την ίδια στιγμή. Ο νέος αυτός τύπος κινητών συσκευών, οι οποίες αναφέρονται ως OpenUEs, έχουν την ικανότητα να διαχωρίζουν το φυσικό τους επίπεδο από τα ανώτερα επίπεδα της στοίβας πρωτοκόλλων. Για να επιτευχθεί κάτι τέτοιο εισάγεται ένα επιπρόσθετο επίπεδο στη στοίβα πρωτοκόλλων, το οποίο καλείται Hypervisor επίπεδο. Το εν λόγω επίπεδο παρεμβάλλεται ανάμεσα στο φυσικό επίπεδο και στα ανώτερά του και εξυπηρετεί τους UEs ώστε να εικονικοποιήσουν τον εαυτό τους. Προκειμένου να μπορεί να λειτουργήσει σωστά το επίπεδο του Hypervisor, εισάγεται μία επιπλέον δυνατότητα στους UEs. Πρόκειται για τη δυνατότητα να δημιουργούν πολλαπλά στιγμιότυπα λογισμικού από το δεύτερο επίπεδο της στοίβας πρωτοκόλλων

τους και πάνω. Κάθε τέτοιο στιγμιότυπο ονομάζεται εικονικό UE (virtual UE - VUE), και έχει τη δυνατότητα να προσομοιώνει τη λειτουργικότητα ενός πραγματικού UE ο οποίος θα μπορεί να συνδέεται ταυτόχρονα σε διαφορετικά δίκτυα. Επίσης, απαιτείται και εδώ η ύπαρξη SDN Agents και SDN Server. Η σχηματική αναπαράσταση της μορφής της αρχιτεκτονικής που προτείνεται φαίνεται παρακάτω.



Εικόνα 2 : Η προτεινόμενη αρχιτεκτονική (device-centric)

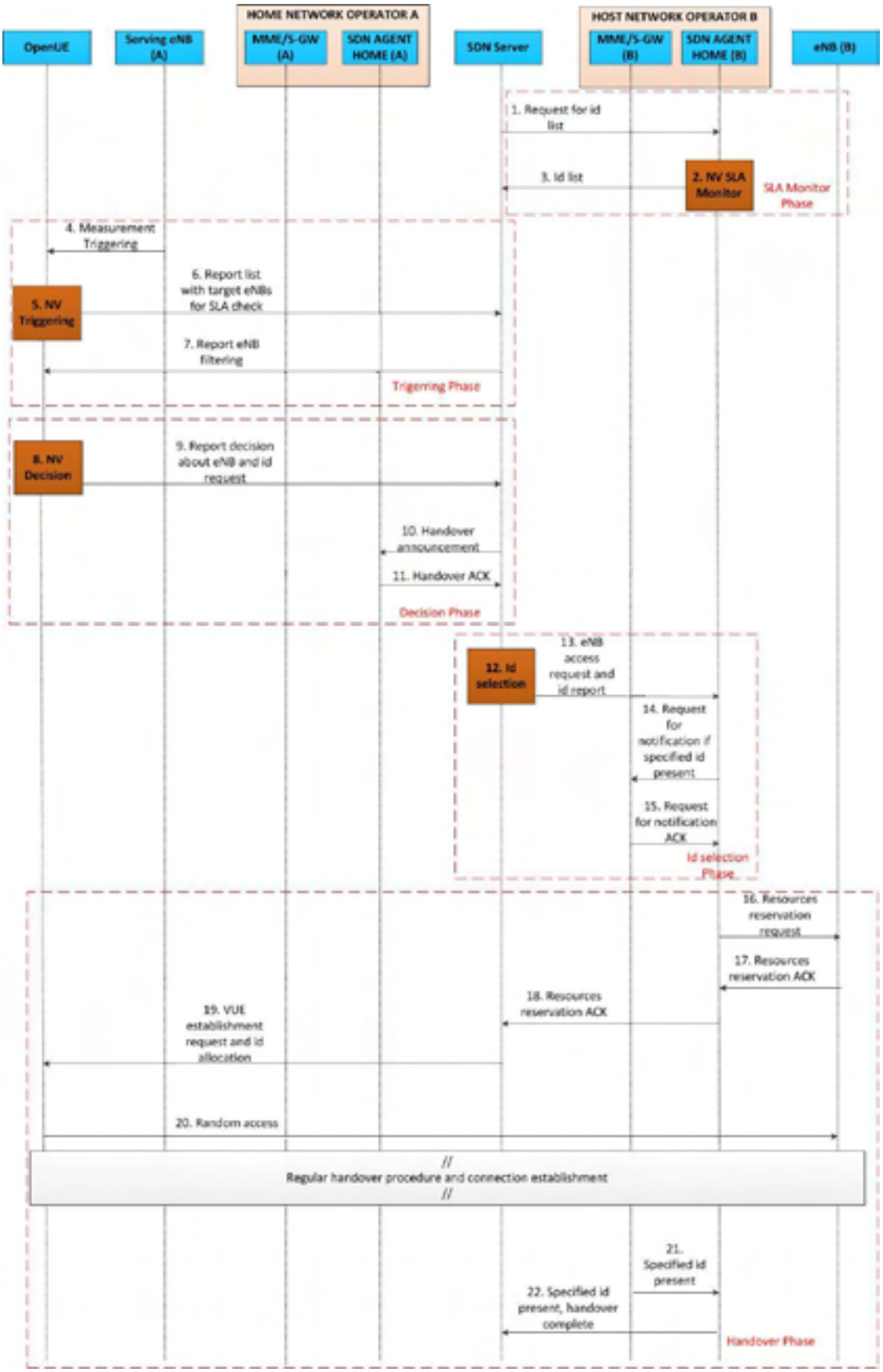
Για την υποστήριξη ενός device-centric πρωτοκόλλου προσδιορίστηκε η απαραίτητη ροή σηματοδοσίας. Η εν λόγω σηματοδοσία ξεκινά να πραγματοποιείται για πρώτη φορά όταν μία κινητή συσκευή που είναι εγγεγραμμένη σε ένα δικτυακό πάροχο Α και εξυπηρετείται από έναν eNB που ανήκει σε αυτόν κρίνει απαραίτητο να αποκτήσει σύνδεση με κάποιον eNB ο οποίος ανήκει σε έναν άλλο δικτυακό πάροχο Β. Στη device-centric προσέγγιση, κυρίαρχης σημασίας είναι ο ρόλος του SDN Server, ο οποίος αποτελεί το σημείο αναφοράς για την υλοποίηση της υπηρεσίας RRS και λόγω της συνολικής εικόνας των επιμέρους δικτύων από τα οποία αποτελείται το δίκτυο, διαθέτει αυξημένες αρμοδιότητες. Μία από τις κύριες αρμοδιότητές του είναι η εκχώρηση ενός μοναδικού αναγνωριστικού-ταυτότητας (id) στην κινητή συσκευή όταν αυτή εισέρχεται σε ξένο δίκτυο,

επιλέγοντας από ένα σύνολο id που διαθέτει για κάθε διαφορετική περιοχή της υποδομής του δικτύου (Tracking Area -TA).

Ως πρώτο βήμα της προτεινόμενης διαδικασίας σηματοδοσίας μέσα στο δίκτυο, SDN Server ζητάει από τον SDN Agent κάθε δικτύου να του αποστείλει μία λίστα με ids τα οποία διαθέτει για εκχώρηση σε νέες κινητές συσκευές που εισέρχονται στο δίκτυό του. Διατηρώντας τις εν λόγω λίστες, ο SDN Server είναι πλέον σε θέση να αναλαμβάνει εκείνος την εκχώρηση ids στα UEs αντί για τον εκάστοτε SDN Agent κάθε φορά (βήμα 1). Ως απόκριση στο αίτημα του SDN Server, ο κάθε SDN Agent πραγματοποιεί διαδικασία NV SLA Monitor ώστε να ελέγξει τις συμφωνίες που έχει συνάψει με άλλους παρόχους (βήμα 2) και στη συνέχεια αποστέλλει τη λίστα με τα ids (βήμα 3). Όταν κριθεί απαραίτητο, ο eNB του οικιακού δικτύου που εξυπηρετεί την κινητή συσκευή ζητά από αυτή να πραγματοποιήσει ένα σύνολο από προσδιορισμένες φυσικές μετρήσεις, στέλνοντάς της ένα μήνυμα «Measurement Triggering» (βήμα 4). Η κινητή συσκευή στη συνέχεια προχωράει σε διαδικασία μετρήσεων και με βάση τα αποτελέσματα αυτών αποφασίζει εάν χρειάζεται να δημιουργήσει σύνδεση με κάποιον άλλο σταθμό βάσης (βήμα 5). Εάν μία τέτοια σύνδεση κριθεί απαραίτητη, η κινητή συσκευή αποστέλλει μία λίστα με όλους τους eNBs που κρίνει κατάλληλους για να την υποδεχθούν στον SDN Server (βήμα 6), από τον οποίο ζητάει να ελέγξει με ποιους από τους εν λόγω eNBs υπάρχει κάποια SLA με το δίκτυο Α στο οποίο ανήκει η συσκευή. Ως απάντηση στο αίτημα της συσκευής, ο SDN Server προχωράει σε διαδικασία filtering (φιλτραρίσματος) και αποστέλλει τη λίστα των eNBs με τους οποίους ο πάροχος δικτύου Α στον οποίο είναι εγγεγραμμένη η κινητή συσκευή έχει υπογεγραμμένη κάποια SLA (βήμα 7). Η εν λόγω λίστα είναι δυνατόν να επιστρέφεται σε μορφή ενός πίνακα στον οποίο θα είναι καταγεγραμμένοι όλοι οι eNBs τους οποίους ανέφερε η κινητή συσκευή στο βήμα 6 καθώς και μία ένδειξη αντίστοιχη στον καθένα από αυτούς (για παράδειγμα ένα bit το οποίο θα παίρνει τις τιμές 0 ή 1) η οποία θα υποδεικνύει εάν υπάρχει ή όχι κάποια SLA μεταξύ του παρόχου Α και του παρόχου στον οποίο ανήκει ο κάθε eNB. Η κινητή συσκευή στη συνέχεια, διαθέτοντας τις πληροφορίες σχετικά με τις SLAs είναι σε θέση να επιλέξει από την αρχική λίστα την οποία έστειλε στον SDN Server στο βήμα 6 τον eNB ο οποίος είναι καταλληλότερος για εκείνη και του οποίου το δίκτυο έχει υπογεγραμμένη συμφωνία με το δίκτυο της συσκευής. Η εν λόγω φάση λήψης της απόφασης για τον κατάλληλο eNB ονομάζεται NV Decision Phase (βήμα 8). Μετά τη λήψη της απόφασης για τον eNB που θα φιλοξενήσει το κινητό, ο UE αναφέρει την απόφασή του στον

SDN Server γνωστοποιώντας του τον επιλεγμένο eNB και ζητώντας του ένα προσωρινό id με το οποίο θα εισέλθει στο νέο δίκτυο (βήμα 9). Στο σημείο αυτό ο SDN Server ενημερώνει τον SDN Agent Home για το γεγονός ότι πρόκειται να πραγματοποιηθεί μεταπομπή (βήμα 10) και εκείνος αποκρίνεται αντίστοιχα (βήμα 11). Σημειώνεται οτι ο SDN Agent Home δεν γνώριζε τίποτα για τη διαδικασία μεταπομπής της συσκευής και ενημερώθηκε μόνο λίγο πριν αυτή πραγματοποιηθεί. Δηλαδή το οικιακό δίκτυο διατηρείται εκτός της διαδικασίας του HO και απλώς λαμβάνει μία ενημέρωση την κατάλληλη χρονική στιγμή. Στη συνέχεια, ο SDN Server προχωράει στη φάση Id Selection (βήμα 12), δηλαδή επιλέγει το id που θα εκχωρηθεί στην κινητή συσκευή και κατόπιν στέλνει στον SDN Agent Host αίτημα πρόσβασης στον επιλεγμένο eNB το οποίο περιλαμβάνει το id που επιλέχθηκε για την κινητή συσκευή και την ταυτότητα του eNB που προσδιορίστηκε ως κατάλληλος (βήμα 13).

Πριν πραγματοποιηθεί αίτημα για τη δέσμευση ραδιοπόρων στον eNB, ο SDN Agent Host ζητάει από την MME του δικτύου του να τον ειδοποιήσει σε περίπτωση που εκείνη εντοπίσει το προαναφερθέν id στο δίκτυο (βήμα 14) και εκείνη αποκρίνεται με ένα μήνυμα επιβεβαίωσης (βήμα 15). Στη συνέχεια και αφού οι πόροι δεσμευτούν ο eNB αποστέλλει μήνυμα επιβεβαίωσης της δέσμευσής τους στον SDN Agent Host (βήμα 17) το οποίο προωθείται και στον SDN Server (βήμα 18). Εκείνος με τη σειρά του στέλνει στην κινητή συσκευή ένα αίτημα δημιουργίας ενός εικονικού στιγμιοτύπου VUE του εαυτού της ώστε να δημιουργηθεί η νέα σύνδεση, ενώ παράλληλα της γνωστοποιεί και το id το οποίο της έχει εκχωρηθεί για τη σύνδεση στο δίκτυο B (βήμα 19). Στο σημείο αυτό πραγματοποιείται το αίτημα για σύνδεση από τον UE στον eNB του ξένου δικτύου, το οποίο περιλαμβάνει και το UE id (βήμα 20) που είναι απαραίτητο στον eNB ώστε να αναγνωρίσει τον UE ως δικό του UE, δηλαδή εγγεγραμμένο στο δίκτυό του. Στη συνέχεια ακολουθεί η συνήθης διαδικασία μεταπομπής που ακολουθείται στα σημερινά δίκτυα κινητών επικοινωνιών προκειμένου να εγκατασταθεί η νέα σύνδεση μεταξύ του UE και του νέου eNB. Τα τελευταία βήματα στη ροή σηματοδοσίας της device-centric προσέγγισης ύστερα από την εγκαθίδρυση της σύνδεσης είναι η ενημέρωση του SDN Agent Host από την αντίστοιχη MME ότι εντοπίστηκε στο δίκτυο κινητή συσκευή με το δεδομένο id (βήμα 21), ο οποίος με τη σειρά του προωθεί την ειδοποίηση στον SDN Server και τον ενημερώνει ότι η διαδικασία της μεταπομπής έχει ολοκληρωθεί (βήμα 22). Η προτεινόμενη ροή σηματοδοσίας απεικονίζεται σχηματικά παρακάτω.

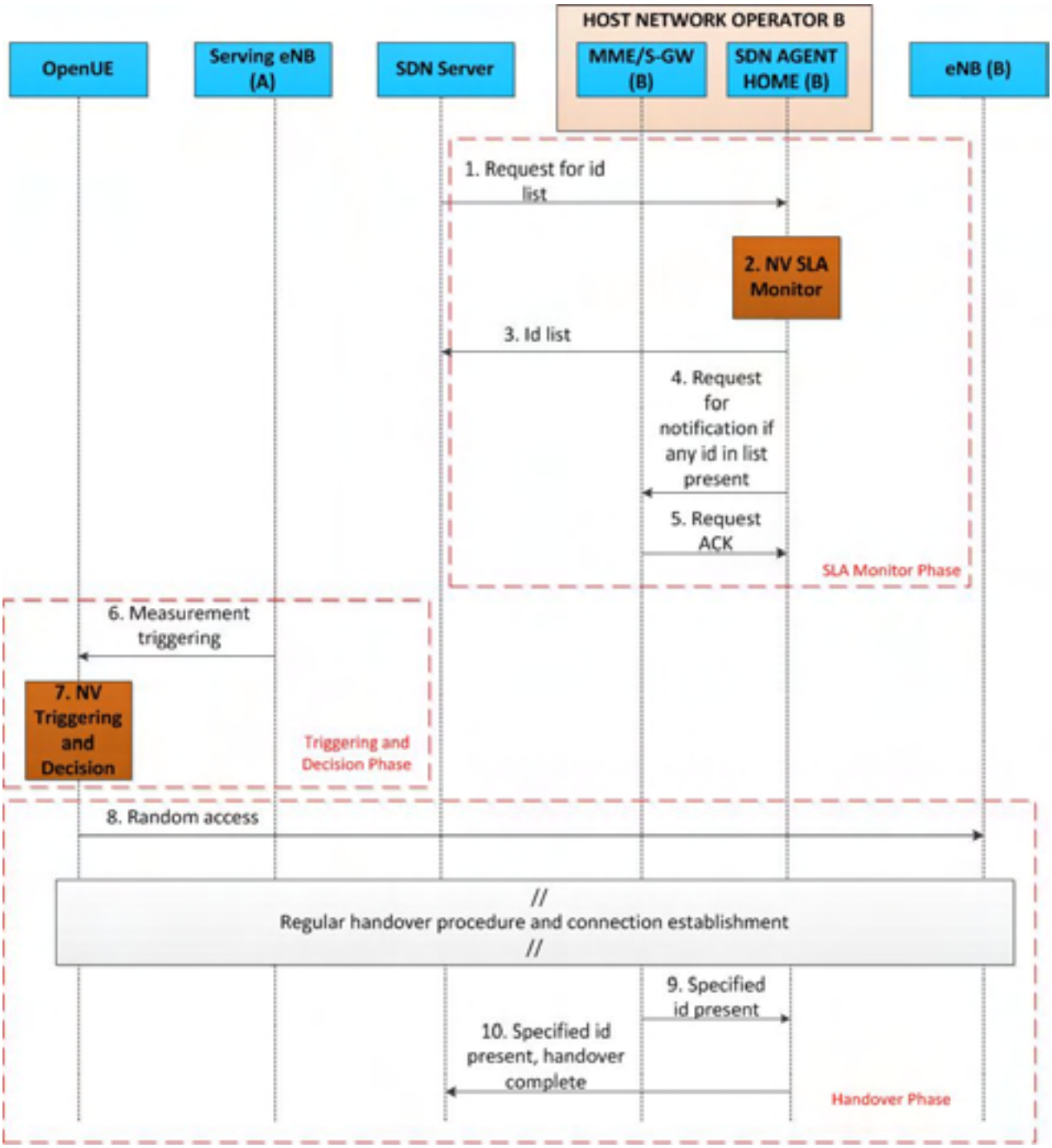


Εικόνα 3 : Το προτεινόμενο πρωτόκολλο διαμοιρασμού ραδιοπόρων

Η προτεινόμενη ροή σηματοδοσίας, όπως περιγράφηκε, καλύπτει όλες τις παραμέτρους οι οποίες πρέπει να ληφθούν υπόψη. Όσον αφορά στις διαδικασίες χρέωσης, αυτές θα πραγματοποιούνται στον SDN Server, ο οποίος έχοντας λάβει την ενημέρωση από τον SDN Agent Host ότι εντοπίστηκε στο δίκτυό του το id που εκχωρήθηκε στον UE θα πιστοποιεί την πραγματοποίηση της μεταπομπής και θα μπορεί να χρεώνει τόσο τους παρόχους όσο και το συνδρομητή δίκαια και χωρίς κινδύνους παρατυπιών. Δεδομένου όμως ότι τα ζητήματα ασφαλείας, χρονισμού και κατανάλωσης ενέργειας τίθενται στο επίκεντρο των 5G συστημάτων, κρίνεται απαραίτητη η επανεξέταση της προτεινόμενης λύσης προκειμένου να εφαρμοστούν βελτιώσεις προς ικανοποίηση ενδεχόμενων κενών στους προαναφερθέντες τομείς.

Οι εν λόγω βελτιώσεις παρατηρούνται και στις πέντε φάσεις που σημειώνονται στα τετράγωνα με κόκκινο περίγραμμα στην Εικόνα 3.2. Πρωτίστως, έχουν αφαιρεθεί τα βήματα ενημέρωσης του SDN Agent Home από τον SDN Server, σχετικά με το επικείμενο handover της κινητής συσκευής (βήματα 10 και 11 στην Εικόνα 3.2). Η αφαίρεση των εν λόγω βημάτων συνίσταται στο γεγονός ότι αφενός η ύπαρξη τους δεν προσφέρει κάποια επιπρόσθετη λειτουργικότητα στον προτεινόμενο μηχανισμό, καθώς το οικιακό δίκτυο δεν είναι απαραίτητο να ενημερώνεται για τα handovers μιας συσκευής προς ξένα δίκτυα, αν βέβαια θεωρήσουμε τον SDN Server ως μία πλήρως έμπιστη οντότητα που θα διαχειριστεί αντικειμενικά και με ακεραιότητα τις επικείμενες χρεώσεις για τις υπηρεσίες που θα χρησιμοποίηση η κινητή συσκευή ως μέλος του ξένου δικτύου. Ο SDN Agent Home, μάλιστα, είναι δυνατόν να μην υπάρχει καν στο οικιακό δίκτυο κορμού, κάτι που σημαίνει ότι το οικιακό δίκτυο ακόμη κι αν δεν χρησιμοποιεί SDN τεχνικές δεν εμποδίζει τα OpenUEs από το να συνδεθούν σε ξένα δίκτυα. Επιπλέον, στα πλαίσια της νέας προτεινόμενης λύσης, έχουν αφαιρεθεί τα βήματα 13,16,17,18,19 της Εικόνας 3.2, που αφορούν την αποστολή αιτήματων πρόσβασης προς το επιλεγμένο eNB και την δέσμευση ραδιοπόρων πριν τη δημιουργία του VUE. Η αφαίρεση των εν λόγω από το μηχανισμό, τον φέρνει πιο κοντά στη διαδικασία μεταπομπής, όπως αυτή πραγματοποιείται μεταξύ σταθών βάσης του ίδιου παρόχου, ενέχοντας ωστόσο σε κάποιες περιπτώσεις τον κίνδυνο δημιουργίας του VUE, αλλά τελικά την μη ολοκλήρωση της μεταπομπής, λόγω ενδεχόμενης έλλειψης πόρων στον επιλεγμένο eNB. Τέλος, προτείνεται η εισαγωγή μίας επιπρόσθετης δικλείδας ασφαλείας σχετικά με τη συχνότητα κατά την οποία ο ίδιος ο OpenUE μπορεί να προβαίνει σε διαδικασίες handover, αλλά και τον μέγιστο αριθμό παράλληλων συνδέσεων σε

διαφορετικά δίκτυα, κατά τις οποίες ένας OpenUE κάνει χρήση πολλών διαφορετικών ids. Με βάση τις προτεινόμενες αλλαγές, η βελτιωμένη ροή σηματοδοσίας διαμορφώνεται όπως φαίνεται παρακάτω.



Εικόνα 4 : Το βελτιστοποιημένο προτεινόμενο πρωτόκολλο RRS

4. Συγκριτική Ανάλυση Προσεγγίσεων και Συμπεράσματα

Έχοντας περιγράψει τις network-centric και device-centric προσεγγίσεις για την υλοποίηση της υπηρεσίας RRS, παρακάτω συνοψίζονται σε πίνακα η συγκριτική ανάλυση μεταξύ τους.

Προσέγγιση	Πλεονεκτήματα	Μειονεκτήματα
Network-centric	• Λιγότερα κενά ασφαλείας • Μεγαλύτερη διάρκεια ζωής μπαταρίας • Μεγαλύτερη ευκολία στη διαδικασία της χρέωσης, αφού ο VeNB «έρχεται» στο οικιακό δίκτυο • Η κινητή συσκευή δεν αντιλαμβάνεται τις διαδικασίες για την υλοποίηση του RRS	• Η εγκαθίδρυση του tunnel δεσμεύει περισσότερους δικτυακούς πόρους • Αυξημένη σηματοδοσία και συνεπώς αυξημένος φόρτος στο δίκτυο • Αύξηση CAPEX και OPEX για τους παρόχους, λόγω της λήψης αποφάσεων μέσα στο δίκτυο
Device-centric	• Οι έξυπνες κινητές συσκευές προσαρμόζονται στο δίκτυο • Λήψη των αποφάσεων από τις συσκευές, άρα μειωμένη σηματοδοσία εντός του δικτύου • Μικρότερος χρόνος λήψης αποφάσεων • Δυνατότητα πολλών παράλληλων συνδέσεων • Διαχωρισμός uplink και downlink κίνησης • Μείωση CAPEX και OPEX για τους παρόχους	• Περισσότερα κενά ασφαλείας • Μικρότερος χρόνος ζωής της μπαταρίας • Η πραγματοποίηση της χρέωσης πρέπει να γίνει από κάποια τρίτη οντότητα

Εικόνα 5 : Συγκριτική ανάλυση network-centric πρωτοκόλλου με το προτεινόμενο πρωτόκολλο

Συνοψίζοντας, συμπεραίνουμε ότι η πορεία προς τα δίκτυα πέμπτης γενιάς είναι αναπόφευκτη, καθώς η χρήση κινητών συσκευών αλλά και η ζήτηση σε υπηρεσίες πολυμεσικών εφαρμογών όπως το video θα προκαλέσουν έκρηξη στην κίνηση δεδομένων και συνεπώς στο φόρτο των σημερινών δικτύων κινητών επικοινωνιών.

Στα πλαίσια της ανάπτυξης των δικτύων 5G εγείρεται και η ανάγκη για όσο το δυνατόν καλύτερη αξιοποίηση των δικτυακών πόρων. Εστιάζοντας σε αυτή την κατεύθυνση, μελετάται πλέον ο σχεδιασμός υπηρεσίας διαμοιρασμού των ραδιοπόρων μεταξύ των διαφορετικών δικτυακών παρόχων. Με το προτεινό- μενο πρωτόκολλο διαμοιρασμού μειώνεται δραστικά η επιβάρυνση του δικτύου, καθώς όλες οι απαιτούμενες ενέργειες πραγματοποιούνται από το κινητό. Ακόμα σημαντικότερα όμως, δίνεται πλέον η δυνατότητα για εγκαθίδρυση ταυτόχρονων παράλληλων συνδέσεων μεταξύ της κινητής συσκευής και περισσότερων του ενός σταθμών βάσης, αφού με τη χρήση τεχνικών εικονικοποίησης το κινητό μπορεί να δημιουργεί εικονικά στιγμιότυπα του εαυτού του. Έτσι, η συνολική κίνηση μίας συσκευής μπορεί να διοχετευθεί και να εξυπηρετηθεί από διαφορετικούς σταθμούς βάσης, εξισορροπώντας το δικτυακό φόρτο.

Αναφορές

[1] M.Yang, et.al, “OpenRAN: A Software-defined RAN Architecture via Virtualization”, SIGCOMM, Hong Kong, August 2013.

[2] K. Pentikousis et al., “MobileFlow: Toward Software-Defined Mobile Networks,” IEEE Communications Magazine (Volume 51, Issue 7), Jul 2013.

[3] J.S. Panchal et al., “Mobile Network Resource Sharing Options: Performance Comparisons,” IEEE Transactions on Wireless Communications, (Volume 12, Issue 9), September 2013.

[4] D. Xenakis et. al., “Radio Resource Sharing as a Service in 5G : A Software Defined Networking Approach”.

High-dimensional visual similarity search: k-d Generalized Randomized Forests

Georgios Samaras (gsamaras@di.uoa.gr)

Abstract

We propose a new data-structure, the generalized randomized k-d forest, or k-d GeRaF, for approximate nearest neighbor searching in high dimensions. In particular, we introduce new randomization techniques to specify a set of independently constructed trees where search is performed simultaneously, hence increasing accuracy. We omit backtracking, and we optimize distance computations, thus accelerating queries. We release public domain software GeRaF and we compare it to existing implementations of state-of-the-art methods including BBD-trees, Locality Sensitive Hashing, randomized k-d forests (implemented in FLANN), and product quantization. Experimental results on image and geometric data, mainly SIFT and GIST visual descriptors and handwritten digits (MNIST), indicate that our method would be the method of choice in dimensions around 1,000, and probably up to 10,000, and datasets of cardinality up to a few hundred thousands or even one million; this range of inputs is encountered in image matching and searching today. For instance, we handle a real dataset of 10^6 GIST images represented in 960 dimensions with a query time of less than 1 sec on average and 90% responses being true nearest neighbors.

Keywords: Randomized Tree, Space Partition, Image Search, High Dimension, Open Software, GIST Images

Advisor

Ioannis Emiris, Professor

1. Introduction

After a couple of decades of work, Nearest Neighbor Search remains a fundamental optimization problem with both theoretical and practical open issues today, in particular for large datasets in dimension well above 100. An exact solution using close to linear space and sublinear query time is impossible, hence the importance of approximate search, abbreviated NNS. We focus on the Euclidean metric but extensions to other metrics should be possible.

Definition 1. Given a finite dataset $X \subset \mathbb{R}^d$ and real $\varepsilon>0$, $x^* \in X$ is an ε-approximate nearest neighbor of query $q \in \mathbb{R}^d$, if $dist(q, x^*) \leq (1 + \varepsilon)dist(q, x)$ for all $x \in X$. For $\varepsilon = 0$, this reduces to exact NNS.

Despite a number of sophisticated methods available, it is still open which is best for various ranges of the input parameters. Here, we propose a practical data-structure, generalizing k-d trees, which should be the method of choice in dimension roughly in the range of 1,000 to 10,000 and inputs of a few hundred thousand points, and up to a million. By taking advantage of randomization and new algorithmic ideas, we offer a very competitive open software for (approximate) NNS in this range of inputs, which provides a good trade-off between accuracy and speed. Our work also sheds light into the efficiency of k-d trees, which is one of the most common data structures but whose complexity analysis is far from tight.

High-dimensional NNS arises naturally when complex objects are represented by vectors of d scalar features. NNS tends to be one of the most computationally expensive parts of many algorithms in a variety of applications, including computer vision, pattern recognition and classification, multimedia databases, data compression, knowledge discovery and data mining, machine learning, document retrieval and statistics [4, 8, 9, 12, 13, 16, 18, 20, 22, 23]. Large scale problems are quite common in such areas, for instance more than 10^7 points and more than 10^5 dimensions [17].

Previous work. There are many efficient approaches to NNS. We focus on the most competitive ones, with emphasis on practical performance, in particular for applications in image similarity search.

An important class of methods consists in data-dependent methods, where the decisions taken for space partitioning are based on the given data points. The Balanced Box Decomposition (BBD) tree [5] is a variant of the quadtree, most closely related to the fair-split tree and the k-d tree. It has $O(\log n)$ height, and subdivides space into axis-aligned hyper-rectangles, containing one or more points with bounded aspect ratio. It achieves query time $O(d^{d+1} \log n/\varepsilon^d)$, using space in $O(dn)$, and preprocessing time in $O(dn \log n)$. The implementation in library ANN[1] seems to be the most competitive method for the NNS problem, for roughly $d<100$. Recently, a novel dimensionality reduction method has been combined with BBD-trees to yield NNS with optimal space requirements and sublinear query time [1].

The performance of BBD-trees in practice is comparable to that of k-d trees. The latter lack a tight analysis but it is known that search becomes almost linear in n for large d because of backtracking. Randomization is a powerful idea: in [19], a random isometry is used with k-d trees; in [21], tree height is analyzed under random rotations; Random Projection trees [10] take another track. R-trees and their variants are most frequent in database applications: they are comparable in performance to k-d trees, but lack complexity and error bounds.

k-d trees are probably the most common data-structure for NNS, having implementations in libraries ANN, with performance comparable to BBD-trees, and CGAL, which is competitive only for small inputs. A successful contribution has been library FLANN [15,16], considered state-of-the-art for d about 100; the method has been most successful on SIFT image descriptors with d =128. FLANN[2] constructs a forest of up to 6 randomized k-d trees and performs simultaneous search in all trees. It chooses the split coordinates adaptively but all leaves contain a single point. The implementation adopts some optimization techniques, such as unrolling the loop of distance computation, but our software goes significantly further in this direction.

In high dimensional space, tree-based data structures are affected by the curse of dimensionality, i.e., either the running time or the space requirement grows exponentially in d. An important method conceived for high dimensional data

1 *http://cs.umd.edu/~mount/ANN*

2 *http://cs.ubc.ca/research/flann*

is Locality Sensitive Hashing (LSH). LSH induces a data independent space partition and is dynamic, since it supports insertions and deletions. The basic idea of LSH is to hash the points of the data set so as to ensure that the probability of collision is much higher for objects that are close to each other than for those that are far apart. The existence of such hash functions depends on the metric space. In general, LSH requires roughly $O(dn^{1+\rho})$ space and $O(dn^{\rho})$ query time for some $\rho \in (0, 1)$. It is known [3] that in the Euclidean case, it is possible to bound ρ by $\rho \leq 1/(1+\varepsilon)^2$. One implementation that we use for comparisons is in library E^2LSH^3 [2].

A different hashing approach is to represent points by short binary codes to approximate and accelerate distance computations. Recent research on learning such codes from data distributions is very active [22]. A more general approach is to use any discrete representation of points, again learned from data points. A popular approach is product quantization (PQ) [12], which both compresses data points and provides for fast asymmetric distance computations, where points remain compressed but queries are not. A powerful non-exhaustive search method inspired by PQ is the inverted multi-index [7]. A combination of such ideas recently led to a very efficient method for clustering large image sets [6]. There are several recent extensions, and the current state of the art in up to 10^9 points in 128 dimensions is locally optimized product quantization (LOPQ) [13].

Contribution. Our main contribution is to propose a new, randomized data-structure for NNS, namely the k-d Generalized Randomized Forest (k-d GeRaF), which generalizes the k-d tree in order to perform fast and accurate NNS in high dimensions (e.g. 1000) and dataset cardinality in thousands or millions. Our main motivation is image datasets, in particular GIST images, as well as image patches of handwritten digits (MNIST dataset). We employ adaptive and randomized algorithms for choosing the split coordinate, and further randomization techniques to build a number of independent k-d trees. We also provide automatic configuration of the parameters governing tree construction and search. All trees are searched simultaneously, with no need for backtracking. The number of trees depends on the input and may go up to the tens or hundreds. We examine alternative ideas, such as random shuffling of the points, random isometries,

3 *http://www.mit.edu/~andoni/LSH*

leaves with several points, and methods for accelerating distance computation. By keeping track of encountered points, we avoid repeated computations [16]. We analyze the theoretical and practical aspects of our approach with emphasis on the experimental analysis for image data.

We have implemented all of the above techniques within a public domain C++ software, GeRaF. This has also allowed us to experiment with different alternatives and provide a simple yet effective automatic parameter configuration. We compare to the main existing alternative libraries on a number of synthetic and real datasets of varying dimensionality and cardinality, including SIFT and GIST images. We have experimented with parameters of all methods and observed the difficulty, in general, to optimize them. Automatic configuration, at least on image data illustrated in this paper, works very satisfactorily for GeRaF, which is the fastest method in building the required space partitions. GeRaF also scales very well, even for $d = 10^4$ or $n = 10^6$, and, at the same accuracy, it is faster than competition for d roughly in the range (10^3, 10^4), and n in the hundreds of thousands or millions, as is the case of modern high-scale image processing and computer vision applications.

Contents. The paper is structured as follows. Section 2 discusses the data structure and method, including randomization factors, building the forest, searching, and improvements that we introduce. Section 3 focuses on more technical implementation issues. Section 4 presents experimental evaluation and comparisons, while conclusions are drawn in Section 5.

2. The k -d GeRaF

The limitations of a single k-d tree for high d are overcome by searching multiple, randomized trees, simultaneously. This section discusses randomization, and algorithms for parameter configuration, building, and searching. Overall, m different randomized k-d trees are built, each with a different structure such that search in the different trees is independent; i.e., neighboring points that are split by a hyperplane in one, are not split in another. Search is simultaneous in the m trees, i.e., nodes from all trees are visited in an order determined by a

shared priority queue. There is no backtracking, and search terminates when c leaves are visited.

2.1. Randomization

The key insight is to construct substantially different trees, by randomization. Multiple independent searches are subse-quently performed, increasing the probability of finding approximate nearest neighbors. Randomization amounts to either generating a different randomly transformed pointset per tree (e.g., rotation or shuffling), or choosing splits at random at each node (e.g., split dimension or value). As discussed below, we investigate four randomization factors, which we use either independently or in combination:

1. Randomly rotate the pointset, before the building process.
2. Randomly choose a cutting dimension.
3. Add a random factor to the cutting value.
4. Random shuffling of point indices.

Rotation. For each k-d tree, we randomly rotate the input pointset or, more generally, apply a different isometry [19]. Each resulting tree is thus based on a different set of dimensions. Only the transformation matrix R is stored for each tree, and not the rotated set. In fact, not even the entire matrix needs to be stored, as discussed in section 2.2. During search, the query is rotated using R before descending each tree. However, distances are computed between the original stored points and the original query.

Split dimension. In a conventional k-d tree, the pointset is halved at each node along one dimension; dimensions are examined in order even for high d. Here, we find the t dimensions of highest variance for the input set and then choose uniformly at random one of these t dimensions at each node. Thus, different trees are built from the given pointset.

Split value. The default split value in a conventional k-d tree is the median of the coordinates in the selected split dimension. FLANN uses the mean for reasons of speed. Here, we compute the median, which would yield a perfect tree, and then randomly perturb it [18]. In particular, the split value δ equals the median plus a quantity uniformly distributed in $[3\Delta / \sqrt{d}, \beta\Delta / \sqrt{d}]$, where Δ is the diameter of the current pointset; δ is computed at every node during building [21].

Shuffling. When computing the split value at each node in a conventional k-d tree, the current pointset at the node is used, which is a subset of the original point-set. Even if the split value is randomized, it is still possible that the same point is chosen if the same coordinate value occurs more than once in the selected dimension. This is particularly common when points are quantized; for instance, SIFT vectors are typically represented by one byte per element. We thus randomly shuffle points at each tree. Hence, different splits occur despite ties.

2.2. Building

The overall building algorithm for k-d GeRaF, consisting of m trees, is outlined in Algorithm 1. For simplicity, only the random split dimensions are included, while the split value is the standard median. There is a random data transformation f per tree, which may include either an isometry, shuffling, or both; in case of an isometry, it is stored for use during search.

Given a dataset X, the t dimensions of maximum variance, say D, are computed. For each tree, X is transformed according to a different function f and then the tree is built recursively. At each node, one dimension (coordinate), say s, is chosen uniformly at random from D and X is split at the median in s. The two subsets of X, say L, R, are then recursively given as input datasets to the two children of the node. The split node so constructed contains the split dimension s and the split value v. Splitting terminates when fewer than p points are found in the dataset, in which case the point indices are just stored in a leaf node. When n is much higher than d, the bottleneck of the algorithm is finding the median, which is $O(n)$ on average. Otherwise, the bottleneck is computing the variance per dimension, which is $O(d)$. The space requirement for the entire data structure

is $O(nd)$ for the data points and $O(nm)$ for the trees, including both nodes and indices to points, for a total of $O(n(d + m))$.

Algorithm 1: k-d GeRaF: building

```
 input: pointset X, #trees m, #split-dimensions t, max #points per
leaf p
 output: randomized k -d forest F
1  begin
2     V ← <VARIANCE of X in every dimension>
3     D ← <t dimensions of maximum variance V>
4     F ← ∅                                                  > forest
5     for i ← 1 to m do
6          f ← <random transformation>                       > isometry, shuffling
7          F ← F ⋃ [ (f, BUILD(f(X)))                        > build on transformed X, store f
8     return F
9  function BUILD(X)                                        > recursively build tree (node/leaf)
10    if |X| ≤ p then                                       > termination reached
11         return leaf(X)
12    else                                                  > split points and recurse
13         s ← <one of dimensions D at random>
14         v ← <MEDIAN of X in dimension s>
15         (L, R) ← <SPLIT of X in dimension s at value v>
16         return node(c, v, BUILD(L), BUILD(R))            > build children on L, R
```

Algorithm 2: k-d GeRaF: searching

```
 input: query point q, forest F , #neighbors k, max #leaf-checks c
 output: k nearest points
1  begin
2     Q.INIT()                                              > min-priority queue, initially empty
3     for i ← 1 to m do
4          DESCEND(q, F[i], FALSE)                          > descend i-th tree, store path in Q, no checks
5     ℓ ← 0                                                 > # of leaves checked
6     H:INIT(k)                                             > min-heap of size k
```

```
7     while ¬ Q.EMPTY()∧ ℓ <c/(1 + ϵ) do
8           (N, d) ← Q.EXTRACT-MIN()                    > (node, distance)
9           DESCEND(q, N, true)                         > descend again, but check leaves now
10          ℓ ← ℓ + 1                                   > increase leaves checked
11    return H
12 function DESCEND(q, node N, check)                   > descend node N for query q
13    d ← N.DIST(q)                                     > signed distance to boundary
14    if d < 0 then                                     > q is in negative half-space
15          Q.INSERT(N.right, |d|)                      > remember right child
16          DESCEND(q, N.left, check)                   > descend left child
17    else
18          Q.INSERT(N.left, |d|)                       > and vice versa
19          DESCEND(q, N.right, check)
20 function DESCEND(q, leaf N, check)                   > test query q on leaf N
21    if ¬ check then return;
22    for i ∈ N.POINTS do
23          H.INSERT(i, ||q - X_i||²)                   > distances to points X_i in leaf N
```

Each random isometry can be a rotation [21] or reflection, and in general requires the generation of a random orthogonal matrix R. We rather use an elementary Householder reflector P for efficiency [19]. In particular, given unit vector $u \in \mathbb{R}^d$ normal to hyperplane H, the orthogonal projection of a point x onto H is $x - (u^\top x)u$. Its reflection across H is twice as far from x in the same direction, that is, $y = x - 2(u^\top x)u = Px$, where $P = I - 2uu^\top$. Although P is orthogonal, the computation of reflection Px is $O(n)$, involving a dot product and an element-wise multiplication and addition. This is because $uu^\top$ is of rank one. We only need to store vector u for each tree.

2.3. Searching

Searching takes place in parallel in all trees; this does not refer to independent search per tree, but rather that nodes from all trees are visited in a particular order using a shared min-priority queue Q. The idea is that given a bound c on the total leaves to be checked, the query iteratively descends the most promising nodes from all trees, and the criterion is the distance of the query to the hyper-

plane specified by each node.

A shown in Algorithm 2, the query initially descends all trees of forest F while all visited nodes are stored in Q, without checking any leaves. Then, for each node extracted from Q, the query descends again, this time computing distances to all points in the leaf. For each decision made at a node while descending, the other one is stored in Q. In particular, the signed distance $d = N.DIST(q)$ of query q to the hyperplane specified by node N is:

$$N.DIST(q) = N.tree.f(q)_{N.c} - N.v \quad (1)$$

where $N.tree.f$ is the isometry of the tree where N belongs, and $N.c$, $N.v$ are the split dimension (coordinate) and value of N, respectively. One child of N is chosen to descend according to the sign of d, and the other is stored in Q with the absolute distance $|d|$ as key. This key is used for priority in Q.

Results are stored in a min-heap H that holds up to k points, where k is the number of neighbors to be returned. For each leaf visited, the distance between q and all points stored in the leaf is computed. For each point X_i of the dataset X, H is updated dynamically such that it always contains the k nearest neighbors to q. The key used for H is the computed (squared) distance $||q - X_i||^2$. A separate array keeps track of points encountered so far, such that no distance is computed twice; this detail is not shown in Alg. 2.

For each tree built under isometry f, the transformed query $f(q)$ is used in all tests at internal nodes, but the initial query q is rather used in all distance computations with points stored at leaves. Similarly, the transformed dataset is used only for building the tree but is not stored. This is possible since the isometry leaves distances unaffected. In practice, unlike (1), the query is transformed according to isometries of all trees prior to descending.

Although no backtracking occurs, visiting new nodes is an implicit form of backtracking. However, given the bound on the number of leaves to be visited, search is approximate. In particular, apart from the case when Q is empty, search terminates when $c / (1 + \varepsilon)$ leaves have been checked. That is, up to c leaves are checked for $\varepsilon = 0$, while this bound decreases for $\varepsilon > 0$, making search faster and less accurate.

3. Implementation

This section discusses our C++ implementation of k-d GeRaF[4], which is available online. The project is open source, under the BSD 2-clause license. The code has been compiled with g++ 4.8 compiler, with several flags enabled, e.g., optimization flags, related to vectorization and loop unrolling. Our code is designed so as to allow the compiler to optimize it. It uses advanced features of C++11, such as **std::thread**. It contains about 4,000 lines of code. Important implementation issues are discussed here, focusing on efficiency.

Parameters. Our implementation provides several parameters to allow the user to fully customize the data structure and search algorithm:

- s Number of points used for computing variance: using a subset of the points accelerates building. Our experiments show this does not a ect accuracy.
- m Number of trees in forest. A small number yields fast building and search, but may reduce accuracy; a large m covers space better and enhances accuracy, but slows down building and search.
- t Number of dimensions used for splits. As d increases, a larger t is better, until accuracy begins to drop. The optimum t depends on the input.
- p Maximum number of points per leaf. A large p means short trees, and saves space; a small p accelerates search, but may reduce accuracy.
- c Maximum number of leaves to be checked during search. The higher this number, the higher the accuracy and search time.
- ε Determines search accuracy (Definition 1); more accurate search comes at the expense of slower query.
- k Number of neighbors to be returned for a query; specified during search.
- δ Random value added to median to define split value. This factor does not help much, thus disabled by default, since it is computationally costly (as shown in table 4).

4 *submitted as supplementary material*

shu Defines whether to perform random shuffling or not.

rot Defines whether to apply a random isometry or not. Isometry is not particularly helpful, thus disabled by default, since it is computationally costly (as shown in table 3).

We provide the same number of parameters in automatic configuration as FLANN, namely ε and k. In the case of manual setting, we provide more, thus offering the possibility of full customization to the user. Moreover, the effective range of parameters often differ with FLANN, since our construction is different.

Configuration. We provide a simple and fast automatic configuration method for parameter tuning. Given a dataset and ε we automatically configure all parameters above, except k. In particular, taking into account n, d and the five coordinates of greatest variance, we configure parameters $p,\ c,\ t,\ m$, limiting their values to powers of two. The particular values chosen are piecewise constant functions of $\varepsilon,\ n,\ d,$ where constants have been obtained by experience, i.e. by manually setting parameters on a number of datasets. This kind of tuning is largely subjective. The runtime is negligible, since the variances are computed by the algorithm anyway. However, the resulting parameter set is not optimal, e.g. in terms of accuracy or speed.

Tree structure. Every tree consists of split nodes and leaves. A split node contains the split dimension and value, while a leaf contains a number of point indices. Points are stored only once, regardless of forest size. We store trees in arrays to benefit from contiguous storage. As discussed in section 4, split value randomization is not beneficial so we disable it. In this case, we split at medial and trees are perfect, thus space is optimized. No re-allocation is needed because we know the size of the tree in advance.

Algorithm 3: Modified Knuth's online variance algorithm

```
input: sequence x of real vectors in ℝ^d
output: variance on each dimension of the vectors in x
1 begin
```

```
2     if x.SIZE() < 2 then return return 0;          > zero vector in ℝ^d
3     μ ← 0; v ← 0                                   > zero vector in ℝ^d
4     for n ← 1 to x:SIZE() do
5           α ← 1/n                                  > α: scalar
6           δ ← x[i] - μ                             > δ: vector in ℝ^d
7           μ ← μ + αδ                               > μ: vector in ℝ^d
8           v ← v + δ∘(x[i] - μ)                     > ∘: Hadamard product; v: vector in ℝ^d
9  return v/(n - 1)
```

Median, variances. The median is found efficiently by the quickselect algorithm, with average complexity $O(n)$. Variance is computed by an extension of Knuth's online algorithm [14, p.232], as shown in Algorithm 3. In particular, we extend the algorithm to operate in parallel on a sequence of vectors rather than scalars. In doing so, we replace vector division with scalar n by multiplication with $\alpha = 1/n$. This choice provides significant speed-up.

Distance computation. This is the most expensive task during search in high dimensions. To speed it up, we note that squared Euclidean distance between point x and query q is $\|q - x\|^2 = \|q\|^2 + \|x\|^2 - 2q^\top x$, where $\|q\|$ is constant, while $\|x\|$ can be stored for all points. Thus distance computation reduces to dot product, providing a speed-up of > 10% in certain cases, as shown in table 1. The space overhead is one scalar per point, which is negligible in high dimensions since all points are stored in memory.

Parallelization. The building process is trivially parallelizable: we just assign building of individual trees to different threads, making sure that the work is balanced among threads. In table 2 we present how much faster the building process gets with parallelization enabled for a small sized forest. Searching is not performed in parallel: due to use of a single priority queue for all trees, more work would be required for communication between different threads. It would be interesting to investigate this extension in future work.

4. Experiments

This section presents our experimental results and comparisons on a number of synthetic and real datasets. All experiments are conducted on a processor at 2.40 GHz x 4 with 3.8 GB memory, except for GIST dataset with n = 106, for which we use a processor at 3 GHz 4 with 8 GB. We compare to BBD-trees as implemented in ANN, LSH as implemented in E^2LSH, FLANN, and our implementation of PQ.

Table 1: Time spent for computing $\|point - query\|^2$, for d = 128.

Approach	time (sec)
online computation	3.06
stored sums	2.67

Table 2: Build with and without parallelization on SIFT data.

n	d	parallel	sec
10.000	128	no	0.015
10.000	128	yes	0.009
1.000.000	128	no	3.32
1.000.000	128	yes	1.4

Table 3 : Build times for n = 10^4,d = 10^4.

Rotation	build (sec)
yes	1.35
no	0.26

Table 4 : Build times for n = 10^3; d = 10^4. Approach `no diam' means that no δ factor was used..

Approach	build (sec)
no diam	0.06
appr diam	1.92
exact diam	89.8

Table 5: Build time (s) for three representative datasets. FLANN does not finish after 4 hr, which is indicated by `-' on Klein bottle or build times in gray on SIFT, where we have skipped configuration and used default values. BBD runs out of memory on SIFT, as well as LSH for ε = 0, 0.1.

	Sphere n = 10^3; d = 10^4				Klein n = 10^4; d = 10^2				MNIST n = 60k; d = 784				SIFT n = 10^6; d = 128			
ε	0	0.1	0.5	0.9	0	0.1	0.5	0.9	0	0.1	0.5	0.9	0	0.1	0.5	0.9
BBD	1.25	1.26	1.30	1.25	0.13	0.14	0.17	0.14	187.5	184.3	185.1	185.6	-	-	-	-
LSH	0.21	0.16	0.18	0.31	0.11	0.07	0.03	0.05	1.47	69.76	48.47	14.35	-	-	170.1	145.5
FLANN	25.0	25.4	25.5	25.6	-	-	-	-	244.	217.2	157.3	142.0	20	19.2	19.8	19.7
GeRaF	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.08	8.167	8.567	8.579	8.565	62.6	93.6	90.5	96.0

Datasets. We use five datasets of varying dimensionality and cardinality. To test special topologies, the first two, Klein bottle and Sphere are synthetic. We generate points on a Klein bottle and a sphere embedded in $\mathbb{R}^d$, then add to each coordinate zero-mean Gaussian noise of standard deviation 0.05 and 0.1 respectively. In both cases, queries are nearly equidistant to all points, which implies high miss rates.

The other three datasets, MNIST[5], SIFT and GIST[6] [12], are common in computer vision, image processing, and machine learning. MNIST contains vectors of 784 dimensions, that are 28x28 image patches of handwritten digits. There is a set of 60k vectors, plus an additional set of 10k vectors that we use as queries. SIFT is a 128-dimensional vector that describes a local image patch by histograms of local gradient orientations. GIST is a 960-dimensional vector that describes globally an entire image. SIFT and GIST datasets each contain one million vectors and an additional set for queries, that are 10^4 for SIFT and 1000 for GIST. For GIST, we also use the first 10^5 vectors as a separate smaller dataset.

Parameters. Most experiments use the default parameters provided by exist-

5 *http://yann.lecun.com/exdb/mnist/*

6 *http://corpus-texmex.irisa.fr/*

ing implementations but, on specific inputs, we have optimized the parameters manually. This improves performance, but is quite impractical in general. FLANN and GeRaF determine automatically the parameters given the dataset and ε, while ANN uses default parameters regardless of ε. E^{2}LSH provides automatic parameter configuration, but not for the most important one, R, used in solving a randomized version of R-near neighbor. This is a major drawback, since the user has to manually identify R at every input. As discussed below, accuracy measurements only refer to the first nearest neighbor, so we always set k = 1 in Alg. 2. The same holds for BBD and FLANN, but not for LSH where the number of neighbors is only controlled by R.

In k-d GeRaF, we have observed that rotation does not seem to affect search performance, despite the time penalty, i.e. build time increases from 0.26 to 1.35 sec on Klein bottle with n = 10^4; d = 10^4. Similarly, split value randomization brings no benefit, despite its cost: build time increases from 0.06 to 1.92 (89.8) sec for approximate (exact) diameter computation, while search accuracy decreases for approximate computation. We have therefore disabled these two randomization factors.

Implementation. Before presenting experimental comparisons to other methods, we measure the effect of two implemen-tation issues discussed in section 3, in particular parallelization and distance computation. Both are measured on SIFT dataset with d = 128. On four cores, parallelization reduces build time from 15 to 9msec for n = 10^4, and from 3.32 to 1.48 sec for n = 10^6: the speedup is higher for larger forests. On the other hand, reduction of distance computation to dot product reduces build time from 3.06 to 2.67 μsec per point. However, this approach appears to be effective only when d > 100 in practice.

Preprocessing. For all methods this includes building, but for FLANN and GeRaF it also includes automatic parameter configuration. Build time is related to the required precision as expressed by ε. For LSH, ε is failure probability and its build time is the most sensitive to ε. Despite requesting the user to manually determine parameter R, LSH performs an automatic parameter configuration as well, which is included in the building process.

Table 5 shows representative experiments. FLANN has difficulties with automatic configuration, which does not terminate after 4 hr on Klein bottle and is quite slow in general. LSH is unexpectedly fast on MNIST for $\varepsilon = 0$, which may be due to the parameters chosen by auto-tuning. GeRaF works well with automatic configuration, and is typically one order of magnitude faster than other methods. Its preprocessing time may increase with ε since this requires fewer points per leaf, hence more subdivisions.

We additionally carry out an experiment with product quantization (in particular, IVFADC) [12] on SIFT, implemented on Matlab with Yael[7] library. Its off -line processing includes codebook learning, which takes 440 sec for 50 k-means iterations and encoding/indexing, which takes 45 sec. The latter time is competitive if codebooks are existing from a similar dataset, but the total time given a new unknown dataset is quite higher than GeRaF and LSH; and even higher than FLANN with default values.

Table 6: Search accuracy and times for synthetic datasets. Search times in gray represent failure cases where miss rate is 100%. Queries are nearly equidistant to points, which explains high miss rates, especially for BBD and FLANN; `-' indicates preprocessing does not finish after 4 hr

	Sphere n = 10^3; d = 10^4				Klein n = 10^4; d = 10^2				MNIST n = 60k; d = 784			
ε	0	0.1	0.5	0.9	0	0.1	0.5	0.9	0	0.1	0.5	0.9
miss %												
BBD	0	100	100	100	0	59	59	59	1	100	100	100
LSH	45	45	45	45	1	1	20	63	2	2	2	2
FLANN	0	0	0	0	-	-	-	-	100	100	100	100
GeRaF	0	24	24	100	2	3	3	5	2	26	40	81
search (ms)												
BBD	9.100	0.210	0.220	0.200	0.470	0.043	0.046	0.052	12	0.024	0.028	0.026
LSH	17.000	16.000	18.000	17.000	2.700	2.400	1.900	0.850	28.000	24.000	22.000	22.000
FLANN	0.310	0.280	0.350	0.320	-	-	-	-	0.021	0.021	0.020	0.021
GeRaF	0.400	0.200	0.150	0.100	0.100	0.083	0.083	0.070	3.900	2.900	1.500	1.300

7 *https://gforge.inria.fr/projects/yael*

Table 7: Klein bottle search for ε = 0.1, for varying n or d, where the other parameter is fixed. Search times in gray represent failure cases where miss rate is 100%. Queries are nearly equidistant from the points, which explains high miss rates. `-' indicates preprocessing does not finish after 2 hr.

n	d	miss %				search (ms)			
		BB	LSH	FLANN	GeRaF	BB	LSH	FLANN	GeRaF
10^3	100	100	0	16	0	1	212	12	199
	1000	100	50	100	50	5	1850	34	14
	5000	100	0	100	0	39	8675	149	122
	10000	100	37	100	2	276	17000	289	520
1000	10^3	100	50	100	50	5	1850	34	14
10000		100	0	100	0	5	1780	-	390
100000		100	8	100	0	276	-	-	10900

Search. We report query times and miss rates for four representative values of ε. The miss rate is the percentage of queries where the reported neighbor is not the exact one. In case of ties, any point at the same distance as the nearest neighbor is accepted as correct. Table 6 shows results for all methods on three representative synthetic datasets. BBD and FLANN have problems with high miss rate or having failed in automatic preprocessing. LSH is at least one order of magnitude slower than GeRaF. In most cases GeRaF is faster (especially for large d = 10^4), with competitive miss rate, except for FLANN on Sphere with d = 10^4, which is the best dimension for FLANN.

Figure 1 presents four representative datasets with real data, namely SIFT and GIST images. BBD and FLANN have problems, namely they suffer from either running out of memory or not completing automatic-parameters build. GeRaF is typically faster than LSH by at least an order of magnitude at the same accuracy. In all cases, FLANN preprocessing does not terminate after 4 hr so we manually configure parameters because default ones yield even higher miss rates. Ignoring this issue, FLANN is generally the fastest method but with low accuracy. On GIST, with d = 960, GeRaF shows best performance. LSH has 0,5%

better miss rate for n = 10^5, but is quite slower; it also fails with 100% miss rate for n = 10^6. With automatic configuration, GeRaF always yields a good trade-off between accuracy and speed. Note that with $\varepsilon = 0$ there may still be true neighbors missed because of the particular approximate algorithm used.

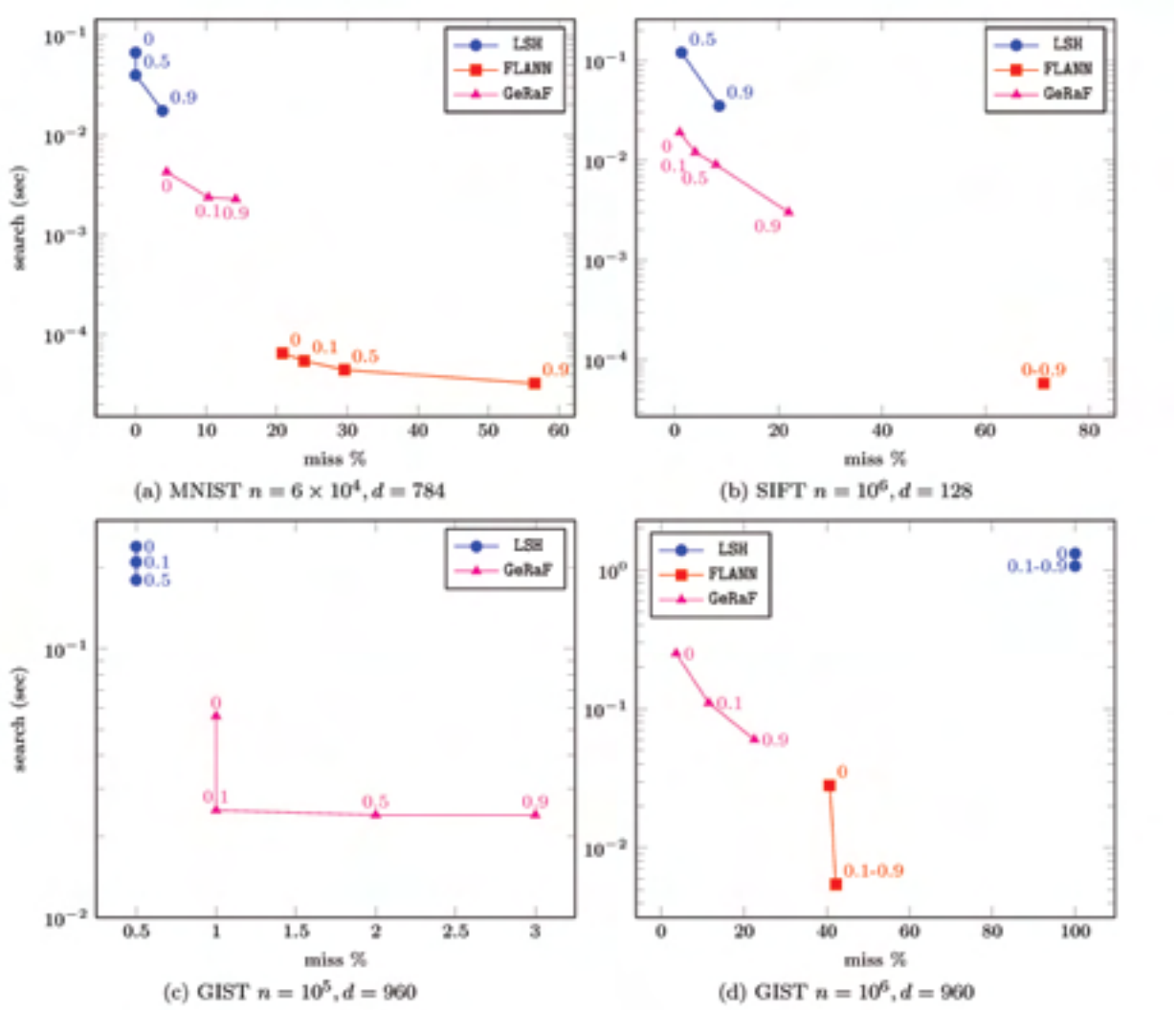


Figure 1: Search accuracy (miss rates) and runtimes (sec) on real datasets. Numbers over points are the values of ε. In (b), LSH is out of memory for ε = {0, 0.1}. In all cases, BBD is out of memory and FLANN does not preprocess after 4 hr for any ε. Its measurements in (a), (b), (d) refer to manually configured parameters.

We also experiment with PQ (IVFADC) in these datasets, which is known to outperform FLANN [12] when combined with re-ranking. For instance, it takes

7 (70) msec for a miss rate of 1% on SIFT (GIST n = 10^6). However its training is slow, as noted above.

Table 7 displays, for all methods, the miss rate and search time as a function of n or d when the other parameter is fixed. In cases where miss rate is not 100%, GeRaF is an order of magnitude faster. The only exception is d = 100, where the situation is inversed with FLANN.

Table 8: GeRaF build and search measurements for Klein bottle dataset with n = 104; d = 102 for varying points per leaf p.

p	256	128	64	32	16	4	2	1
build (s)	0.0592	0.0618	0.0674	0.0695	0.0860	0.1159	0.1543	0.1587
search (ms)	0.2324	0.1863	0.1198	0.0941	0.0712	0.0592	0.0743	0.0928
miss %	1	1	2	7	6	10	14	22

Approximate search evaluation. We also measure for GeRaF the percentage of queries where the reported nearest neighbor does not lie within $1 + \varepsilon$ of the nearest distance. This a more natural measure than miss rate when approximate search is requested given a specific ε. For Klein bottle with n = 10^4, d = 10^2, this rate is 2% and 0%, for ε = 0, and $\varepsilon \in$ {0.1, 0.5, 0.9}, respectively. In order for the output to always lie within 1+ ε of optimal, one may set $c = n$, thus disabling the termination condition of leaves to be checked. However, due to the curse of dimensionality, performance nearly reduces to brute force in this case. For GIST with n = 10^5, d = 960 for instance, search takes 140ms, whereas miss rate is 0% and 0,4% for ε = 0 and $\varepsilon \in$ {0.1, 0.5, 0.9} respectively.

Points per leaf. Finally, we measure the effect of storing multiple points per leaf on the Klein bottle dataset. The results are shown in Table 4. It is clear that search time improves when there are less points per leaf, and this is why a single point per leaf is a common approach. However, the build time and most importantly the miss rate also increase significantly. We therefore provide a reasonable trade-off by automatically adjusting parameter p.

5. Discussion

We have presented an efficient data structure for approximate nearest neighbor search that explores different random-ization strategies, and an efficient implementation, GeRaF, that is found competitive against existing implementations of several state-of-the-art methods. We provide a simple but effective automatic parameter configuration that yields the fastest preprocessing, including both configuration and building, as well as a successful trade-off between accuracy and speed. Most competing methods have difficulties, namely they suffer from running out of memory at large scale (e.g., BBD), slow or non-terminating parameter configuration (e.g., FLANN), or unstable search behavior between accurate (but slow) or fast (but inaccurate) search (e.g., LSH and FLANN). PQ is consistently faster and more accurate at search, but is significantly slower to build, which is impractical when the dataset is updated; PQ is also conceptually harder which im-plies a more complicated implementation. Our findings are consistent on both synthetic and real datasets of a wide range of dimensions and cardinalities, with emphasis on SIFT and GIST images, and image patches representing handwritten digits (MNIST dataset).

An interesting and relevant feature is that GeRaF appears to exploit intrinsic structure in the input, such as the structure of SIFT image datasets or the Klein bottle. The work in [21] may pave the way for explaining this behavior.

Interesting open questions include whether and how GeRaF can be fully dynamic, supporting insertions and deletions, as well as handling batch queries in an optimized manner. Other future directions include performing parallel or distributed search and more principled parameter configuration with discrete optimization. In fact, recent experiments with parameter tuning by genetic algorithms indicate that build time for large datasets such as SIFT can drop by a factor of 100 without significantly affecting search time while reducing miss rate [11].

References

[5] Anagnostopoulos, E., Emiris, I., Psarros, I.: Low-quality dimension reduction and high-dimensional approximate nearest neighbor. In: Proc. Annual Symp. on Computational Geometry, pp. 436{450 (2015)

[6] Andoni, A., Indyk, P.: E2LSH 0.1 User Manual, Implementation of LSH: E2LSH, http:// www.mit.edu/ andoni/LSH (2005)

[7] Andoni, A., Indyk, P.: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. Commun. ACM 51(1), 117{122 (2008). DOI 10.1145/1327452.1327494. URL http://doi.acm.org/10.1145/1327452.1327494

[8] Arandjelovic, R., Zisserman, A.: Extremely low bit-rate nearest neighbor search using a set compression tree. IEEE Trans. on Pattern Analysis and Machine Intell. 36(12), 2396{406 (2014). Doi: 10.1109/TPAMI.2014.2339821

[9] Arya, S., Mount, D., Netanyahu, N., Silverman, R., Wu, A.: An optimal algorithm for approximate nearest neighbors in xed dimension. J.ACM 45, 891{923 (1998). DOI 10.1145/293347.293348. URL http://doi.acm.org/10.1145/293347.293348

[10] Avrithis, Y., Kalantidis, Y., Anagnostopoulos, E., Emiris, I.: Web-scale image clustering revisited. In: Proc. Intern. Conf. Comp. Vision (ICCV), pp. 1502{1510 (2015)

[11] Babenko, A., Lempitsky, V.: The inverted multi-index. In: Computer Vision and Pattern Recognition, pp. 3069{3076. IEEE (2012)

[12] Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classi cation. In: Comp. Vision & Pattern Recognition (2008)

[13] Chum, O., Philbin, J., Isard, M., Zisserman, A.: Scalable near identical image and shot detection. In: Conference on Image and Video Retrieval, pp. 549{556. ACM Press New York, NY, USA (2007)

[14] Dasgupta, S., Freund, Y.: Random projection trees and low dimensional manifolds. In: Proc. ACM Symp. Theory of Computing, pp. 537{546 (2008)

[15] Giachoudis, N.: Report for kd-GeRaF parameter auto tunning. Tech. rep., University of Athens (2015)

[16] J egou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. IEEE Trans. Pattern Analysis & Machine Intell. 33(1), 117{128 (2011)

[17] Kalantidis, Y., Avrithis, Y.: Locally optimized product quantization for approximate nearest neighbor search. In: Comp. Vision & Pattern Recogn. (2014)

[18] Knuth, D.: The Art of Computer Programming, vol. 2. Addison-Wesley (1998)

[19] Muja, M., Lowe, D.: Fast approximate nearest neighbors with automatic algorithm con guration. In: Proc. VISAPP: Intern. Conf. Computer Vision Theory & Appl., pp. 331{340 (2009)

[20] Muja, M., Lowe, D.: Scalable nearest neighbour algorithms for high dimensional data. Pattern Analysis and Machine Intelligence (2014)

[21] Perronnin, F., Akata, Z., Harchaoui, Z., Schmid, C.: Towards good practice in large-scale learning for image classi cation. In: Computer Vision and Pattern Recognition (2012)

[22] Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proc. Computer Vision & Pattern Recogn (2007)

[23] Silpa-Anan, C., Hartley, R.: Optimised kd-trees for fast image descriptor matching. In: Proc. IEEE Computer Vision & Pattern Recognition (2008)

[24] Torralba, A., Fergus, R., Freeman, W.: 80 million tiny images: A large data set for nonparametric object and scene recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence 30(11), 1958{1970 (2008). DOI 10.1109/Tpami.2008.128

[25] Vempala, S.: Randomly-oriented kd-trees adapt to intrinsic dimension. In: Proc. Foundations Software Techn. & Theor. Comp. Science, pp. 48{57 (2012)

[26] Wang, J., Shen, H.T., Song, J., Ji, J.: Hashing for similarity search: A survey. Tech. Rep. 1408.2927, Arxiv (2014)

Σύστημα Συστάσεων για Εστιατόρια: Η Περίπτωση των Εστιατορίων του Λονδίνου

Δωροθέα - Κωνσταντίνα Ε. Τσιμπίδη (dor.tsimpidi@gmail.com)

Περίληψη

Αντικείμενο της παρούσας εργασίας είναι η δημιουργία ενός συστήματος συστάσεων το οποίο αντλεί πληροφορία από ιστοτόπους που περιέχουν αξιολογήσεις χρηστών για τα εστιατόρια του Λονδίνου. Το σύστημα αυτό επιδικιώκει να βελτιώσει τα αποτελέσματα των συστάσεων που εμφανίζονται στην έως τώρα βιβλιογραφία, αξιοποιώντας τα προφίλ των αξιολογητών για να αποδώσει βάρη στις αξιολογήσεις τους, ανάλογα με την αξιοπιστία τους. Το σύστημα απαρτίζεται από διάφορες συνιστώσες οι οποίες πραγματοποιούν προεπεξεργασία δεδομένων από τους επιλεγμένους ιστοτόπους, εξόρυξη πληροφορίας μέσω επεξεργασίας φυσικής γλώσσας, τοποθέτηση ετικετών και παραγωγή των τελικών αποτελεσμάτων με χρήση συντελεστών βαρύτητας. Η τελική εφαρμογή είναι σε θέση να παράγει συστάσεις για τους χρήστες, ανάλογα με τις προτιμήσεις τους.

Λέξεις κλειδιά: Φιλτράρισμα βασισμένο στο Περιεχόμενο, Εξόρυξη Πληροφορίας, Συντελεστές Βαρύτητας, Ετικέτες, Εστιατόρια του Λονδίνου.

Επιβλέπων

Ιζαμπώ Καράλη, Επίκουρη Καθηγήτρια

1. Εισαγωγή

Τα τελευταία χρόνια, ο αριθμός των διαδικτυακών εφαρμογών και καταστημάτων συνεχώς αυξάνεται, δίνοντας τη δυνατότητα στους χρήστες του Διαδικτύου να πλοηγηθούν και να επιλέξουν υπηρεσίες και προϊόντα από μια πληθώρα εναλλακτικών επιλογών. Οι υπηρεσίες και τα προϊόντα αυτά ενδέχεται να είναι σε αριθμό πολύ μεγαλύτερα από εκείνα ενός παραδοσιακού καταστήματος. Καθώς λοιπόν αυξάνονται οι επιλογές που προσφέρονται στους χρήστες, αυξάνεται και ο χρόνος που απαιτείται από αυτούς για να πλοηγηθούν και να επιλέξουν όσα ταιριάζουν στις προτιμήσεις και επιθυμίες τους.

Προκειμένου να αντιμετωπιστεί αυτό το πρόβλημα, οι διάφοροι ιστότοποι εφάρμοσαν τεχνικές εξατομίκευσης (personalization) οι οποίες θα προσαρμόζονταν στις ανάγκες και απαιτήσεις του κάθε χρήστη. Σημαντικό μέρος αυτών των τεχνικών αποτέλεσαν τα Συστήματα Συστάσεων (Recommendation Systems - RS) [1]. Τα συστήματα συστάσεων έχουν γίνει εξαιρετικά δημοφιλή τα τελευταία χρόνια και χρησιμοποιούνται σε μια πληθώρα από εφαρμογές.

Όσον αφορά εφαρμογές στον κλάδο των εστιατορίων, από το 1997 με το σύστημα Entrée [2] έως και σήμερα, έχουν δημιουργηθεί συστήματα συστάσεων στα οποία ο χρήστης δηλώνει τις προτιμήσεις του σε τομείς όπως είναι η ποιότητα του φαγητού και το σύστημα πραγματοποιεί τις ανάλογες συστάσεις. Οι ιστότοπος tripadvisor.com είναι ένα ιδιαίτερα δημοφιλές και ευρέως χρησιμοποιούμενο σύστημα συστάσεων εστιατορίων, το οποίο βασίζει τις συστάσεις του σε προηγούμενες αξιολογήσεις χρηστών. Για τους χρήστες που αξιολογούν τα εκάστοτε εστιατόρια, δημιουργείται προφίλ στο οποίο καταγράφονται στατιστικά του χρήστη, όπως είναι ο αριθμός των αξιολογήσεων που έχει πραγματοποιήσει και οι φορές που οι αξιολογήσεις του είχαν θετική ανταπόκριση από άλλους χρήστες του συστήματος. Ωστόσο, οι υπάρχουσες προσεγγίσεις για την ανάπτυξη τέτοιων συστημάτων δεν έχουν αξιοποιήσει τα προφίλ των χρηστών κατά την αξιολόγηση των εστιατορίων και κατά συνέπεια κατά την εξαγωγή των συστάσεων. Επομένως, η αξιολόγηση ενός χρήστη που θεωρείται αξιόπιστος λαμβάνει την ίδια βαρύτητα με αυτή ενός χρήστη του οποίου οι κριτικές θεωρήθηκαν μη αξιόπιστες από τους υπόλοιπους χρήστες, το οποίο έχει ως αποτέλεσμα οι τελικές αξιολογήσεις και συστάσεις να μην είναι πάντα αξιόπιστες.

Στόχος αυτής της εργασίας ήταν η σχεδίαση και υλοποίηση ενός συστήματος το οποίο διευκολύνει το χρήστη στην επιλογή του καταλληλότερου εστιατορίου με βάση τις προτιμήσεις του. Το σύστημα αναπτύχθηκε λαμβάνοντας υπόψη τα εστιατόρια του Λονδίνου. Ο λόγος ήταν ότι γι'αυτά υπάρχει πολύ πληροφορία διαθέσιμη στο Διαδίκτυο οπότε η επίλυση του προβλήματος γι'αυτά είναι πιο ενδιαφέρουσα και έχει μεγαλύτερη αξία.

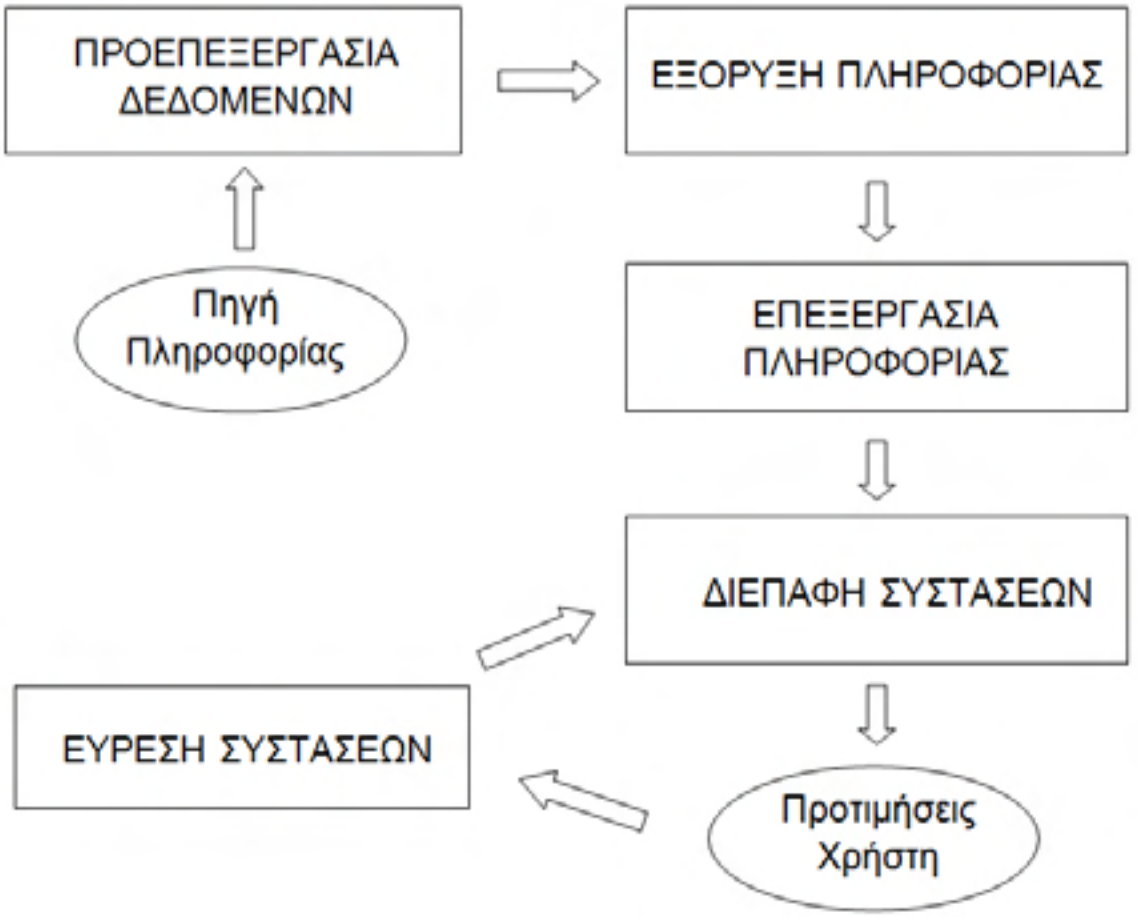
Στο σύστημα αυτό, όσον αφορά τα τελικά συμπεράσματα που παράγονται, λαμβάνονται υπόψιν και τα προφίλ των χρηστών εκ των οποίων προήλθαν οι αξιολογήσεις. Πιο συγκεκριμένα, όσο πιο μεγάλη αποδοχή είχαν οι αξιολογήσεις του κάθε χρήστη και όσο πιο πολλές ήταν σε αριθμό, τόσο πιο μεγάλη ισχύ κατέχει η γνώμη του στην τελική βαθμολογία του εκάστοτε εστιατορίου ή των χαρακτηριστικών του.

Το σύστημα συστάσεων που δημιουργήθηκε είναι σύστημα φιλτραρίσματος βασισμένο στο περιεχόμενο [3]. Αρχικά, εξάγονται κριτικές χρηστών και περαιτέρω γνώση από τέσσερις διαφορετικούς ιστοτόπους με περιεχόμενό τους εστιατόρια της πόλης του Λονδίνου. Κατόπιν, επιλέγονται τα βασικά χαρακτηριστικά των εστιατορίων με βάση τα οποία γίνεται η ομαδοποίηση της πληροφορίας. Ενδεικτικά, η ποιότητα του φαγητού και η εξυπηρέτηση των πελατών αποτελούν μερικά από αυτά τα χαρακτηριστικά. Μέσω της επεξεργασίας κειμένου εξάγονται τα τελικά συμπεράσματα για τα εστιατόρια, τα οποία και αποθηκεύονται. Τέλος, ο χρήστης είναι σε θέση να καταθέσει τις προτιμήσεις του στη διεπαφή του συστήματος συστάσεων, η οποία υπολογίζει τις καταλληλότερες για αυτόν συστάσεις και του εμφανίζει τα αποτελέσματα.

2. Το Σύστημα Συστάσεων

Στη συνέχεια περιγράφεται ένα σύστημα συστάσεων, το «Restaurant Finder», το οποίο δίνει τη δυνατότητα στο χρήστη να ανακαλύψει το καταλληλότερο εστιατόριο του Λονδίνου σύμφωνα με τις προτιμήσεις του.

2.1. Αρχιτεκτονική Συστήματος



Σχήμα 1: Αρχιτεκτονική του συστήματος συστάσεων

3. Μεθοδολογία

3.1. Σύνολο Δεδομένων

Τα δεδομένα που χρησιμοποιούνται προέρχονται από 4 ιστοτόπους, τους tripadvisor.com, yelp.com, hardens.com και zomato.com, που περιλαμβάνουν κριτικές χρηστών και γενικές πληροφορίες για τα εστιατόρια του Λονδίνου. Μέσω αυτών των ιστοτόπων, για κάθε εστιατόριο, γίνεται γνωστή η διεύθυνσή του, οι κουζίνες που προσφέρονται, το μέσο εύρος των τιμών, οι επιλογές που παρέχονται από το εστιατόριο όπως επίσης και κριτικές των χρηστών, η ημε-

ρομηνία που πραγματοποιήθηκε η κάθε κριτική, τα προφίλ των χρηστών και οι θετικές ψήφοι που έλαβε η κριτική.

3.2. Συλλογή και Προεπεξεργασία Δεδομένων

Προκειμένου να συλλεχθούν τα απαραίτητα δεδομένα από τους τέσσερις ιστοτόπους που επιλέχθηκαν, έγινε χρήση ενός προγράμματος ανίχνευσης του Web (Web parser), του Jsoup [4].

Για την πληροφορία που πάρθηκε από αυτούς τους ιστοτόπους, τέθηκε ο περιορισμός ότι για κάθε εστιατόριο κάθε ιστοτόπου, οι κριτικές που θα αποθηκεύονταν σε αυτά δεν θα ήταν παλαιότερες του 2014 και δεν θα ξεπερνούσαν σε αριθμό τις 200. Επίσης, όσες κριτικές δεν ήταν γραμμένες στην αγγλική γλώσσα, απορρίφθηκαν.

3.3. Εξόρυξη Πληροφορίας

Στο πλαίσιο της εργασίας, σκοπός της εξόρυξης πληροφορίας ήταν η συλλογή δομημένης πληροφορίας για τέσσερα κριτήρια με βάση τα οποία θα πραγματοποιούνταν οι συστάσεις στο μέλλον. Τα τέσσερα κριτήρια που επιλέχθηκαν ήταν η ποιότητα του φαγητού, η εξυπηρέτηση, η ατμόσφαιρα και η σχέση ποιότητας-τιμής των εστιατορίων. Η επεξεργασία φυσικής γλώσσας πραγματοποιήθηκε μέσω του συστήματος εξόρυξης πληροφορίας ANNIE (a Nearly-New Information Extraction System) που εμπεριέχεται στο Λογισμικό GATE (General Architecture for Text Engineering) [5].

Το ANNIE περιλαμβάνει μια πληθώρα από εξαρτήματα που εκτελούν γραμματική και συντακτική ανάλυση σε κείμενο που δέχονται ως είσοδο. Στα πλαίσια αυτής της εργασίας χρησιμοποιήθηκαν τα ANNIE English Tokenizer, ANNIE Gazetteer, ANNIE Sentence Splitter, ANNIE POS Tagger, JAPE Transducer. Οι συνιστώσες στις οποίες τροποποιήθηκαν οι προεπιλεγμένες ρυθμίσεις ήταν οι ANNIE Gazetteer και JAPE Transducer.

ANNIE Gazetteer:

Ο ρόλος του gazetteer είναι η χρήση λιστών με σκοπό την αναζήτηση οντοτήτων στο κείμενο που δέχεται σαν είσοδο. Στα πλαίσια αυτής της εργασίας

δημιουργήθηκαν 32 λίστες με συνολικά 6171 λέξεις ή φράσεις. Οι λίστες αυτές περιελάμβαναν πιθανές λέξεις που αναφέρονταν στο προσωπικό του εστιατορίου, σε διάφορους τύπους φαγητού, σε λέξεις αναφορικά με τη σχέση ποιότητας-τιμής και σε λέξεις που αναφέρονταν στην ατμόσφαιρα. Δημιουργήθηκαν λίστες με λέξεις που περιγράφουν θετικά, αρνητικά και ουδέτερα συναισθήματα, όπως επίσης και λέξεις πιο εξειδικευμένες που περιγράφουν κάποιο συγκεκριμένο κριτήριο.

JAPE Transducer:

Η γραμματική JAPE δίνει τη δυνατότητα να αναγνωριστούν κανονικές εκφράσεις (regular expressions) σε κείμενα με αναγνωρισμένες οντότητες. Για τη γραμματική JAPE δημιουργήθηκαν 218 κανόνες, με τη βοήθεια των οποίων καθορίζονται και τοποθετούνται οι ετικέτες. Κάποιοι από αυτούς τους κανόνες αφορούν τους τίτλους της κριτικής και κάποιοι άλλοι το κυρίως κείμενο της κριτικής. Η βασική ιδέα ήταν ο εντοπισμός του τρόπου σύνδεσης μεταξύ των λέξεων που υποδηλώνουν την έννοια των χαρακτηριστικών και των λέξεων που φανερώνουν θετικό, μέτριο η αρνητικό συναίσθημα ως προς αυτό.

Ενδεικτικά, μια ετικέτα είναι η εξής: «FOOD_GOOD_ver»

Οι ετικέτες με κατάληξη «ver» (verified-επιβεβαιωμένες) αναφέρονται σε ετικέτες για τις οποίες υπάρχει μεγάλη πιθανότητα να έχουν τοποθετηθεί σωστά και κατά συνέπεια περιγράφουν με ακρίβεια το συναίσθημα που περιγράφει ο χρήστης. Για τις ετικέτες που έχουν κατάληξη «un» (unverified-μη επιβεβαιωμένες), υπάρχει ένα μικρό ενδεχόμενο να μην έχουν τοποθετηθεί σωστά και επομένως να περιγράφουν λανθασμένα το συναίσθημα που περιγράφει ο χρήστης.

3.4. Υπολογισμός Τιμής Χαρακτηριστικών

Μετά την τοποθέτηση των ετικετών, πραγματοποιείται επεξεργασία στο κείμενο προκειμένου να παραχθούν τα τελικά συμπεράσματα για το κάθε εστιατόριο.

Θεωρούμε «Τ» την τιμή στο διάστημα [1-5] για ένα συγκεκριμένο χαρακτηριστικό σε μια συγκεκριμένη κριτική. Για τον υπολογισμό της τιμής Τ λαμβάνονται υπόψη συντελεστές βαρύτητας. Δεδομένου ότι υπάρχουν ετικέτες επιβεβαιωμένες και ετικέτες μη επιβεβαιωμένες, τίθεται στην κάθε περίπτωση διαφορετικός

συντελεστής βαρύτητας.

Κάθε περίπτωση ή υποπερίπτωση για ένα συγκεκριμένο χαρακτηριστικό έχει το δικό του συντελεστή βαρύτητας στον τελικό υπολογισμό της τιμής του χαρακτηριστικού. Θεωρούμε «Β» αυτό το συντελεστή.

Μια κριτική μπορεί να περιέχει χρήσιμη πληροφορία με μορφή κειμένου στις κριτικές, στους τίτλους των κριτικών, στη συνολική βαθμολογία του εστιατορίου και στις βαθμολογίες των χαρακτηριστικών που έχει θέσει κάθε χρήστης. Ανάλογα με την ποσότητα της χρήσιμης πληροφορίας που μπορεί να περιέχει μια κριτική, εμφανίζονται 24 περιπτώσεις για καθένα από τα τέσσερα βασικά χαρακτηριστικά (ποιότητα φαγητού, εξυπηρέτηση, σχέση ποιότητας-τιμής, ατμόσφαιρα), τα οποία έχουν διαφορετικούς συντελεστές «Β» και «Τ».

Ενδεικτικά, για να ισχύσει η Περίπτωση 1 πρέπει να υπάρχει διαθέσιμη βαθμολογία για το χαρακτηριστικό, τίτλος κριτικής με αναφορά στο χαρακτηριστικό και όσες ετικέτες υπάρχουν να είναι επιβεβαιωμένες. Στην Υποπερίπτωση 1.1 επιπλέον ισχύει ότι δεν υπάρχει απόκλιση μεγαλύτερη των 2 (δύο) μονάδων μεταξύ κάποιας επιβεβαιωμένης ετικέτας του χαρακτηριστικού και της βαθμολογίας του χαρακτηριστικού. Για αυτή την υποπερίπτωση, ισχύει ότι

B = 1

T =0,4*βαθμολογία_χαρακτηριστικού + 0,3*τιμή_ετικετών_τίτλου + 0,3*τιμή_επιβεβαιωμένων_ετικετών_κριτικής

Στην Υποπερίπτωση 1.2 υπάρχει απόκλιση μεγαλύτερη των 2 (δύο) μονάδων μεταξύ κάποιας επιβεβαιωμένης ετικέτας του χαρακτηριστικού και της βαθμολογίας του χαρακτηριστικού και η συγκεκριμένη κριτική για το συγκεκριμένο χαρακτηριστικό αγνοείται λόγω αντίφασης.

Επίσης, αξίζει να σημειωθεί ότι ανάλογα με τον αριθμό των αξιολογήσεων και τον αριθμό των αναφορών στα χαρακτηριστικά των εστιατορίων, οι τελικές τιμές διαχωρίστηκαν σε επιβεβαιωμένες (certain) και μη επιβεβαιωμένες (uncertain).

3.5. Υπολογισμός Συντελεστή Βαρύτητας με βάση το Προφίλ του Χρήστη

Θεωρούμε «P» το συντελεστή βαρύτητας της αξιοπιστίας του χρήστη, ο οποίος πολλαπλασιάζεται με το συντελεστή βαρύτητας της κριτικής.

Για τις αξιολογήσεις από τον ιστότοπο tripadvisor.com, όσο πιο πολλές θετικές ψήφους είχε λάβει ο χρήστης για τις κριτικές του, τόσο πιο μεγάλη θεωρήθηκε η αξιοπιστία του. Ο συντελεστής βαρύτητας που αποδόθηκε στον κάθε χρήστη ήταν ο εξής:

P = 1 + θετικές_ψήφοι_κριτικής + συνολικές_θετικοί_ψήφοι_κριτικών/συνολικός_αριθμός_κριτικών_χρήστη

Για τις αξιολογήσεις από τους ιστοτόπους yelp.com και zomato.com, προκειμένου να υπολογιστεί ο συντελεστής αξιοπιστίας του χρήστη, μελετήθηκε η σχέση που ενδέχεται να έχει ο αριθμός των κριτικών του με τον αριθμό των φίλων του στον ιστότοπο. Θεωρήθηκε ότι όσο πιο πολλές κριτικές έγραφε ο χρήστης, αν αυτές ήταν αντικειμενικές και σωστές, τόσο πιο μεγάλος θα έπρεπε να είναι ο αριθμός των φίλων του, καθώς ο ρόλος των «φίλων» στον ιστότοπο επιτρέπει την άμεση πρόσβαση στις κριτικές αυτού του ατόμου. Επίσης, θεωρήθηκε ότι ένας χρήστης με μικρό αριθμό από κριτικές και μικρό αριθμό φίλων ενδέχεται να είναι καινούριο μέλος στον ιστότοπο. Επομένως ο προστιθέμενος συντελεστής θα έπρεπε να είναι μικρός αλλά ταυτόχρονα παρόμοιος με το συντελεστή ενός ατόμου που είχε γράψει μεγάλο αριθμό κριτικών αλλά έχει πολύ μικρό αριθμό φίλων, καθώς αυτό ίσως υποδηλώνει ότι αυτές δεν εξέφραζαν την κοινή γνώμη. Οι μεγαλύτεροι συντελεστές S, όπου το S ήταν παραμετρικό, αποκτήθηκαν από άτομα με μεγάλο αριθμό κριτικών και μεγάλο αριθμό φίλων. Ο συντελεστής βαρύτητας που αποδόθηκε στον κάθε χρήστη ήταν ο εξής:

P = 1 + θετικές_ψήφοι_κριτικής + S

Για τις αξιολογήσεις από τον ιστότοπο hardens.com δεν υπολογίστηκαν συντελεστές αξιοπιστίας των χρηστών λόγω έλλειψης προφίλ των χρηστών.

Με βάση αυτά τα στοιχεία, το σύστημα ήταν σε θέση να υπολογίσει τη βαρύτητα του προφίλ του χρήστη, την αριθμητική τιμή του κάθε χαρακτηριστικού για το

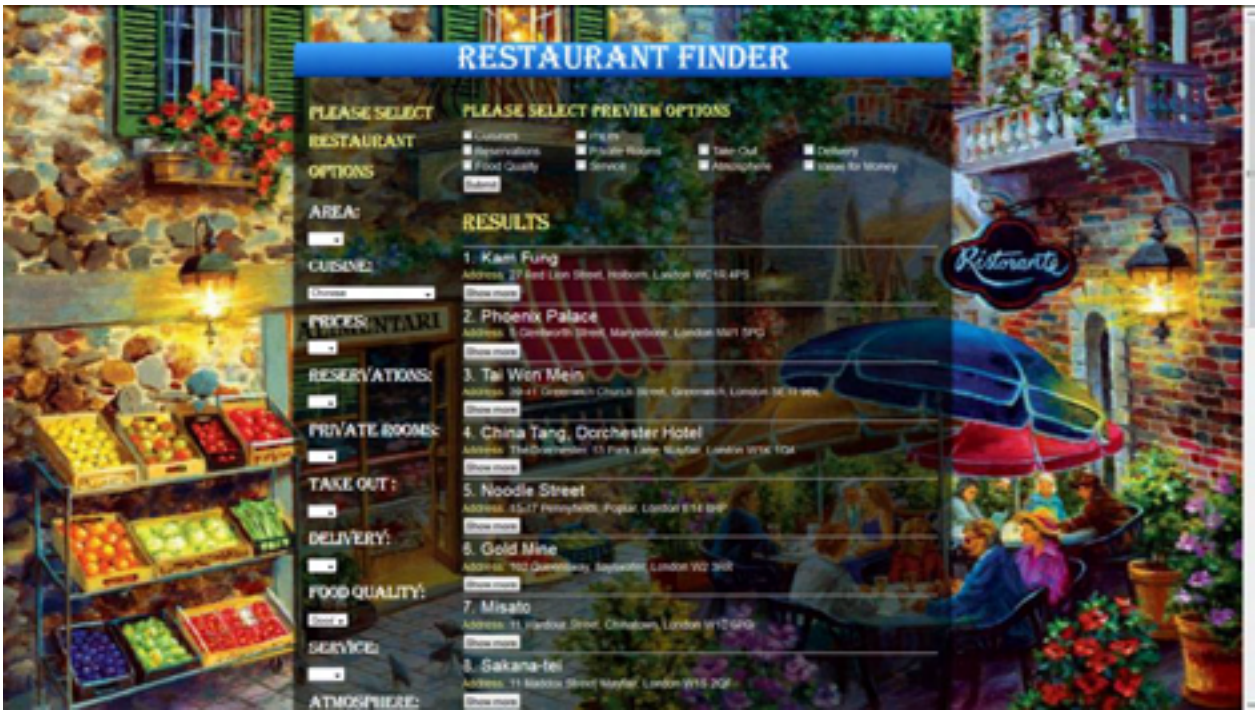
συγκεκριμένο ιστότοπο και τη βεβαιότητα αυτού του χαρακτηριστικού. Η τελική σταθμισμένη τιμή του κάθε χαρακτηριστικού υπολογίστηκε από τη βαρύτητα του προφίλ του χρήστη, την τιμή του χαρακτηριστικού και τους θετικούς ψήφους της κριτικής, εφόσον υπήρχαν.

3.6. Διεπαφή Συστήματος

Τα αποτελέσματα αυτά αποθηκεύτηκαν και έγιναν διαθέσιμα στο χρήστη μέσω μιας διαδικτυακής εφαρμογής με τίτλο «Restaurant Finder». Μέσω της διεπαφής, ο χρήστης είναι σε θέση να καταχωρήσει τις προτιμήσεις του επιλέγοντας από μια πληθώρα επιλογών. Για κάθε πεδίο επιλογών, υπάρχει και η κενή επιλογή η οποία σημαίνει ότι δεν είναι απαραίτητο για το χρήστη να επιλέξει κάποια συγκεκριμένη τιμή για αυτό.

Αρχικά, προσφέρεται ένας μεγάλος αριθμός από επιλογές για το πεδίο της τοποθεσίας, το οποίο αποτελείται από τα 3 πρώτα γράμματα του ταχυδρομικού κώδικα του εστιατορίου. Επίσης, προσφέρονται διαφορετικές κουζίνες από τις οποίες ο χρήστης καλείται να επιλέξει. Στο εύρος των τιμών υπάρχουν 3 επιλογές, από τις πιο οικονομικές έως τις λιγότερο οικονομικές. Όσον αφορά τη δυνατότητα παράδοσης φαγητού στο σπίτι, τη δυνατότητα εστίασης σε ιδιωτικό χώρο του εστιατορίου, τη δυνατότητα επιλογής φαγητού σε πακέτο και τη δυνατότητα κρατήσεων, εκτός από την κενή επιλογή, υπάρχει η επιλογή «Yes». Για τα 4 βασικά χαρακτηριστικά, δηλαδή την ποιότητα του φαγητού, τη σχέση ποιότητας-τιμής, την εξυπηρέτηση και την ατμόσφαιρα, εκτός από την κενή τιμή υπάρχει η επιλογή «Good». Τέλος, δίνεται η επιλογή στο χρήστη να επιλέξει συνδυασμό από τέσσερις ιστοτόπους, τους tripadvisor.com, yelp.com, hardens.com και zomato.com, από τους οποίους θα προέρχονται οι τελικές συστάσεις. Ωστόσο, αν δεν επιλεγεί τουλάχιστον ένας ιστότοπος, το σύστημα συστάσεων λαμβάνει υπόψη και τους τέσσερις ιστοτόπους.

Έστω ότι ο χρήστης έχει θέσει ως κουζίνα την επιλογή «Chinese», έχει θέσει στην ποιότητα του φαγητού και στη σχέση ποιότητας-τιμής την επιλογή «Good», έχει επιλέξει ως ιστοτόπους τους tripadvisor.com και zomato.com και δεν έχει δώσει επιλογή στα υπόλοιπα πεδία. Στην «Εικόνα 1» εμφανίζονται οι συστάσεις για αυτές τις προτιμήσεις.



Εικόνα 1: Αποτελέσματα των συστάσεων

Επιπλέον, για κάθε εστιατόριο υπάρχει η επιλογή «Show more», η οποία εμφανίζει το εκάστοτε εστιατόριο με τα πλήρη στοιχεία, χαρακτηριστικά και παροχές του. Έστω ότι ο χρήστης επέλεξε το εστιατόριο «Kam Fung», στην «Εικόνα 2» παρουσιάζονται τα στοιχεία του.

Εικόνα 2: Πλήρης εμφάνιση στοιχείων εστιατορίου

3.7. Εύρεση Συστάσεων

Όταν ο χρήστης δηλώσει τις προτιμήσεις του, για κάθε εστιατόριο που τηρεί τις προϋποθέσεις που έθεσε, υπολογίζονται οι εξής τιμές:

- **field1:** Η τιμή αυτή αντιπροσωπεύει τον αριθμό των χαρακτηριστικών για τα οποία υπάρχει μη κενή πληροφορία συγκριτικά με τον αριθμό των χαρακτηριστικών για τα οποία ενδιαφέρεται ο χρήστης. Για παράδειγμα, έστω ότι ο χρήστης έχει επιλέξει να δεχτεί πληροφορία από τους ιστοτόπους tripadvisor.com και zomato.com και τα χαρακτηριστικά που τον ενδιαφέρουν είναι η ποιότητα του φαγητού και η σχέση ποιότητας-τιμής. Ο μέγιστος αριθμός που μπορεί να δεχτεί η τιμή «field1» είναι 4, το οποίο αντιπροσωπεύει την ύπαρξη μη κενής τιμής των χαρακτηριστικών «φαγητό» και «σχέση ποιότητας-τιμής» για τον ιστότοπο tripadvisor.com και την ύπαρξη μη κενής τιμής των χαρακτηριστικών «φαγητό» και «σχέση ποιότητας-τιμής» για τον ιστότοπο zomato.com.
- **field2:** Η τιμή αυτή εκφράζει τη μέση βεβαιότητα των χαρακτηριστικών για τα οποία ενδιαφέρεται ο χρήστης.
- **field3:** Εκφράζει τη μέση τιμή των χαρακτηριστικών για τα οποία ενδιαφέρεται ο χρήστης.
- **field4:** Η τιμή αυτή αντιπροσωπεύει τον αριθμό των χαρακτηριστικών για τα οποία δεν έχει δηλώσει ενδιαφέρον ο χρήστης αλλά είναι διαθέσιμα στους ιστοτόπους που επέλεξε.
- **field5:** Η τιμή αυτή εκφράζει τη μέση βεβαιότητα των χαρακτηριστικών για τα οποία δεν έδειξε ενδιαφέρον ο χρήστης αλλά είναι διαθέσιμα στους ιστοτόπους που επέλεξε.
- **field6:** Εκφράζει τη μέση τιμή των χαρακτηριστικών για τα οποία δεν ενδιαφέρεται ο χρήστης αλλά υπάρχουν στους ιστοτόπους που επέλεξε.
- Η μέση τιμή της ποιότητας του φαγητού, η μέση τιμή της σχέσης ποιότητας-τιμής, η μέση τιμή της ατμόσφαιρας και η μέση τιμή της εξυπηρέτησης για τους ιστοτόπους που επέλεξε ο χρήστης.

Όταν υπολογιστούν αυτές οι τιμές για το κάθε εστιατόριο, τα τελικά αποτελέσματα του συστήματος συστάσεων ταξινομούνται με φθίνουσα σει-

ρά σύμφωνα με τις τιμές των «field1», «field2», «field3», «field4», «field5» και «field6», όπου το field1 έχει τη μεγαλύτερη βαρύτητα και το field6 τη μικρότερη.

Αυτή η μέθοδος ταξινόμησης επιτρέπει να εμφανιστούν στην κορυφή της λίστας τα εστιατόρια που περιέχουν την περισσότερη επιβεβαιωμένη θετική πληροφορία για τα χαρακτηριστικά που ενδιαφέρουν το χρήστη, ενώ παράλληλα λαμβάνονται υπόψιν και οι τιμές των υπολοίπων χαρακτηριστικών. Με αυτό τον τρόπο, στην κορυφή των συστάσεων τοποθετούνται τα εστιατόρια που εκτός από τα επιμέρους χαρακτηριστικά, είναι πιο καλά και στο σύνολό τους.

4. Συμπεράσματα

Στην παρούσα πτυχιακή εργασία υλοποιήθηκε ένα σύστημα συστάσεων που εξάγει πληροφορίες και αξιολογήσεις εστιατορίων από 4 ιστοτόπους, τις επεξεργάζεται με τεχνικές επεξεργασίας φυσικής γλώσσας και υπολογίζει τις τελικές βαθμολογίες των χαρακτηριστικών των εστιατορίων, λαμβάνοντας υπόψιν και τα προφίλ των χρηστών που κατέγραψαν τα σχόλια στους εκάστοτε ιστοτόπους.

Η αρχική, μη επεξεργασμένη πληροφορία εξήχθη από 4 ιστοτόπους με περιεχόμενό τους κριτικές εστιατορίων του Λονδίνου. Στις κριτικές αυτές, εκτός από κείμενο, περιεχόταν και αριθμητική αξιολόγηση των βασικότερων χαρακτηριστικών του κάθε εστιατορίου ή του εστιατορίου σαν σύνολο. Ένα σημαντικό προτέρημα του συστήματος είναι το γεγονός ότι πραγματοποιήθηκε επεξεργασία φυσικής γλώσσας στο κείμενο των αξιολογήσεων προκειμένου να εξαχθούν συμπεράσματα για τα 4 βασικότερα χαρακτηριστικά των εστιατορίων. Αυτή η επεξεργασία κρίθηκε απαραίτητη τόσο για τα εστιατόρια στα οποία δεν υπήρχε η επιλογή κατάθεσης βαθμολογίας για τα 4 βασικότερα χαρακτηριστικά, όσο και για τα εστιατόρια όπου υπήρχε βαθμολόγηση αυτών των χαρακτηριστικών, καθώς αποτέλεσε πληροφορία που επιβεβαίωνε τα τελικά αποτελέσματα ή ελλιπή δεδομένα.

Για τη διαδικασία της εξόρυξης πληροφορίας από το κείμενο των αξιολογήσεων δημιουργήθηκαν συνολικά 218 λεξιλογικοί κανόνες και λίστες με 6171 λέξεις ή φράσεις που τους συνόδευαν. Ο μεγάλος αριθμός κανόνων και λέξεων επέτρεψε την τοποθέτηση αρκετά μεγάλου αριθμού ετικετών στις κριτικές των χρηστών. Ωστόσο, δεν ήταν δυνατόν να ληφθούν υπόψιν κανόνες για όλους τους τρόπους λεξιλογικής και συντακτικής έκφρασης που ενδεχομένως να επέλεγε

ο χρήστης για να εκφραστεί στις αξιολογήσεις του. Επίσης, αν και οι κανόνες περιέλαβαν μοτίβα όπου αναγνωρίζονταν οι αρνήσεις σε μια πρόταση, οι οποίες άλλαζαν το νόημα μιας έκφρασης, και σε αυτή την περίπτωση δεν ήταν πάντα δυνατό να εντοπιστούν όλες οι αρνητικές εκφράσεις.

Προκειμένου οι τελικές τιμές των χαρακτηριστικών να θεωρηθούν αξιόπιστες, χρησιμοποιήθηκαν σταθμισμένοι μέσοι όροι για τον υπολογισμό τους. Πιο συγκεκριμένα, κατά τον υπολογισμό των τιμών των χαρακτηριστικών για κάθε εστιατόριο, ανάλογα με το προφίλ του χρήστη που κατέγραψε το σχόλιο και την αποδοχή που έλαβε το σχόλιο, υπολογίστηκε και διαφορετικός συντελεστής βαρύτητας για τον τελικό μέσο όρο του κάθε χαρακτηριστικού. Αυτή η τεχνική συνέβαλε στον υπολογισμό τελικών τιμών χαρακτηριστικών όπου η πιο δημοφιλής άποψη επικράτησε και η λιγότερη δημοφιλής λήφθηκε υπόψη σε μικρότερο βαθμό.

Η διεπαφή της διαδικτυακής εφαρμογής προσφέρει στο χρήστη τη δυνατότητα επιλογής προτιμήσεων από 4 βασικά χαρακτηριστικά αλλά και γενικότερων στοιχείων και παροχών του εστιατορίου, όπως είναι η περιοχή και το εύρος των τιμών σε αυτό. Ένα προτέρημα του συστήματος είναι ότι οι συστάσεις περιλαμβάνουν εστιατόρια ανεξαρτήτως δημοτικότητας. Πιο συγκεκριμένα, καθώς ο αριθμός των αξιολογήσεων δεν λαμβάνεται υπόψη στις τελικές συστάσεις, εστιατόρια τα οποία δεν έχουν μεγάλο αριθμό αξιολογήσεων αλλά οι τιμές των χαρακτηριστικών τους είναι υψηλές και επιβεβαιωμένες, έχουν τη δυνατότητα να βρεθούν στην κορυφή των συστάσεων.

Αναφορές

[1] F. Ricci, L. Rokach and B. Shapira, “Introduction to Recommender Systems Handbook” in Recommender Systems Handbook, Springer, 2011, pp. 1-35

[2] R. Burke, “Hybrid Recommender Systems: Survey and Experiments”, User Modeling and User-Adapted Interaction, vol. 12, pp 331-370, November 2002.

[3] K. Lang, “NewsWeeder: Learning to filter netnews”, Proceedings 12th International Conference on Machine Learning, Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, 1995, pp. 331–339.

[4] J. Hedley, “jsoup: Java HTML Parser”, available at: http://jsoup.org/

[5] Wikipedia, “General Architecture for Text Engineering”, available at: https://en.wikipedia.org/wiki/General_Architecture_for_Text_Engineering

ΔΙΠΛΩΜΑΤΙΚΕΣ ΕΡΓΑΣΙΕΣ

Similarity-based User Identification across Social Networks

Aikaterini Zamani (katerinazam@di.uoa.gr)

Abstract

We study the identifiability of users across social networks, with a trainable combination of different similarity metrics. Motivated by the need to verify information that appears in social networks, we need to identify users across networks. We approach this problem by a combination of similarity measures that take into account the users' affiliation, location, professional interests and past experience, as stated in the different networks. We experimented with a variety of combination approaches, ranging from simple averaging to trained hybrid models. Our experiments show that, under certain conditions, identification is possible with sufficiently high accuracy to support the goal of verification.

Keywords: User Identification, Similarity Learning, Entity Resolution

Advisors

Panagiotis Stamatopoulos, Professor, Georgios Paliouras, Researcher NCSR Demokritos, Dimitrios Vogiatzis, Collaborating Researcher NCSR

1. Introduction

Social network services have became part of our everyday life. It is now commonplace that people have accounts in multiple social networks, sharing their thoughts, promoting their work and probably influencing a part of the population via them. A variety of functionalities are provided by these services, such as video and photo uploading, posting, messaging, republishing etc, differing according to the platform and its aim.

A variety of recent studies focus on the problem of user identification across the web. To the best of our knowledge this is the first study whose motivation is to verify the validity and trustworthiness of information based on public professional information provided by users in social networks. To achieve this, public information from one network can be used to validate the source of information in another network. Therefore, there is a need for user identification across social networks.

In this study, we try to identify users across two popular networks: LinkedIn and Twitter. Our approach relies on novel similarity measures, that mainly take into consideration professional information about the users. To achieve a satisfactory combination of the proposed similarity metrics, we experiment with various supervised classification techniques. In addition, an attempt is made to deal with the imbalanced data problem and estimate the value of missing fields. Experiments based on a real world scenario show the high accuracy in user iden- tification between these networks. Thus, the main contribution of our work is to prove that the proposed approach of combining different similarity metrics is a viable solution to the identification of users, which in turn can be used to verify the validity of public information in social network.

However, many efforts have examined user-account correlation across web profiles by exploiting explicit and implicit information. For example Vosecky, Hong, and Shen [10] combine different explicit profile fields by setting definite comparison vectors. In addition, Iofciu et al. [2] study the influence of tags in user identification across tagging network services relying on the combination of implicit and explicit information. Malhotra et al. [9] utilize explicit feedback, in order to model the digital footprints of users in the Twitter and LinkedIn social networks. Their work is the one that comes closest to our approach, but

it also bears a number of differences from it. Due to our original motivation, we focused on a different set of features to be extracted from the user profiles while we handle differently the problem of imbalanced data. Namely Malhotra et al. [9] use random sub-sampling to balance the training data, thus training their model with the same number of match and mis-match examples. Finally, our work addresses the issue of missing feature values, which is not dealt with in Malhotra et al. [9]

The most recent work of Goga et al. [1] is also the one closer to our work, correlates users across different and popular social networks in large scale. Their study is based on public feature extraction and the proposed similarity metrics deal with explicit information. Due to the large scale of data, they present a classification strategy in order to deal with availability of fields and imbalance.

2. Problem Description

In this study, we focus on individuals that are interested in promoting their professional activities in social media. Thus we experiment on two popular social networks that are used mainly, though not exclusively, for professional purposes: Twitter and LinkedIn. We assume that the individuals often provide their real name in these social networks and therefore, the problem that we need to solve is primarily that of name disambiguation. Specifically, our approach compares users that have similar names, based on public information provided by the users, as returned by search engine of the respective network.

Within a social network, each user is represented by a set of attributes that forms their user profile. We derive a subset of these attributes based on the public accounts of users in the respective network. The LinkedIn profile of a user includes the following attributes: screen name, summary, location, specialization, current/past jobs with the respective affiliations, education, as well as projects and publications. On the other hand, the Twitter profile of a user contains: screen name, short biography, location and the user mentions, that the user specifies in her tweets. Although the process starts with a name search, screen name can be considered as a feature because the results of the search engine do not always fit exactly to the query. Fig. 1 presents a simple example of how the user's attributes are aligned in the two networks, in order to be used

in the similarity metrics.

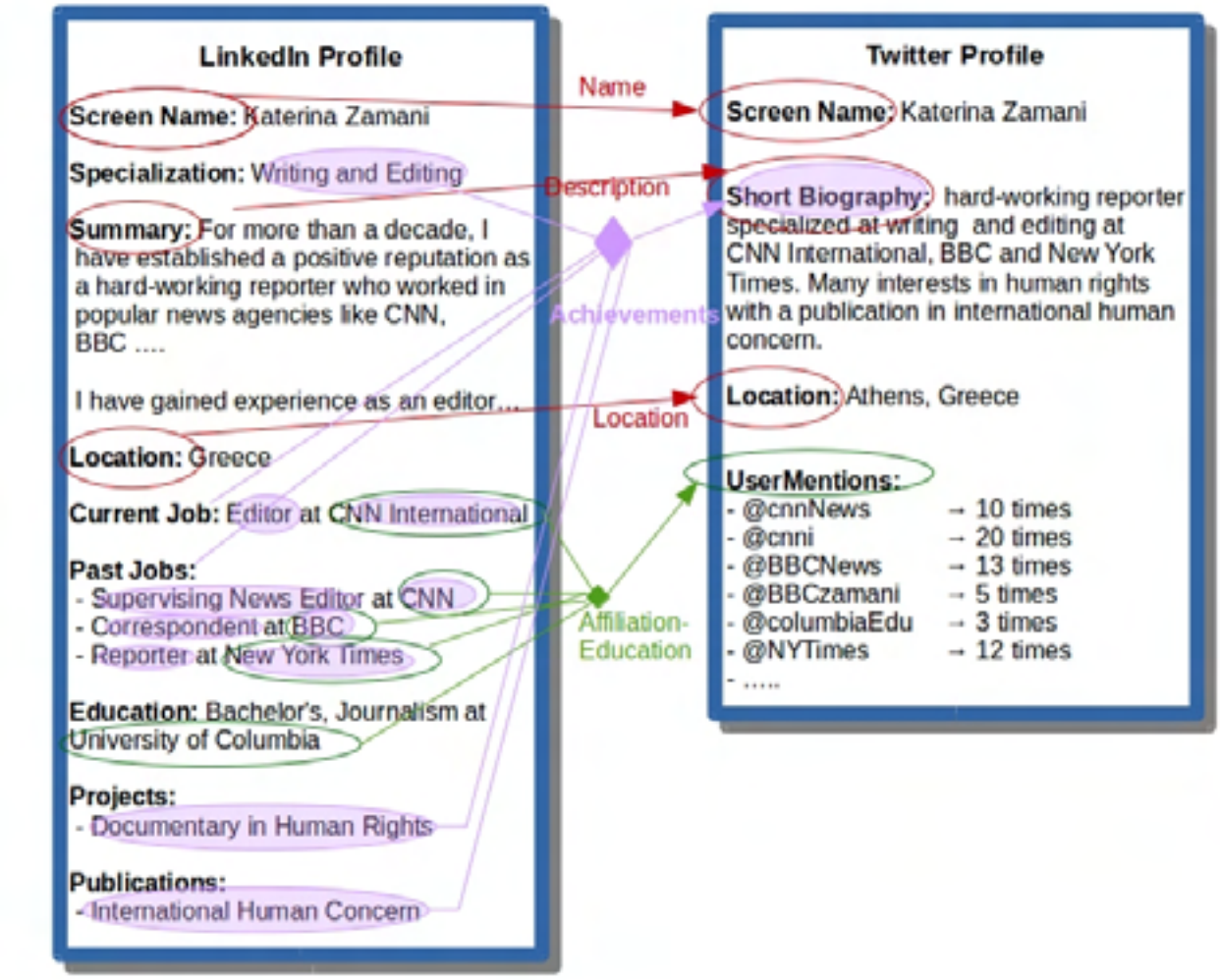


Figure 1: Example profiles and the alignment of their attributes, as used in the similarity metrics.

3. Approach

3.1. Description of Profiles

As explained in the previous section, the basic idea of our approach is to pair accounts that result from name search and identify those that belong to the same user. Therefore, the task that we are dealing with is translated to a classification of account pairs into two classes: “match” and “mis-match”.

Specifically, in order to identify users we create a similarity vector for each pair of users' profiles. The representation of our similarity vector is based on the definition proposed by [10]. Suppose that we have two user profiles from different social networks:

$$u_1 \in SN_1 \quad and \quad u_2 \in SN_2 \tag{1}$$

The similarity vector of the two profiles is defined as:

$$V(u_1, u_2) = < score_1, score_2, \cdots, score_n > \tag{2}$$

where $score_k$ corresponds to the score, returned by the kth similarity metric. In order to facilitate the comparison, the similarity scores are normalized in the range [0.0, 1.0].

3.2. Similarity Measures

In this subsection we describe the similarity metrics that we use, in order to construct the similarity vectors for pairs of user profiles.

Name Measures. Previous work in record linkage [8] recommend Jaro-Winkler as an appropriate similarity for short strings. Therefore, in our approach we use the Jaro-Winkler distance in order to find the similarity between the screen names of users –first and last name that a user provides during her registration.

Description Measures. The basic idea is inspired from the fact that users often provide common phrases in their description in different social networks. This measure estimates the similarity between the short biographies or summaries that users provide in different social networks, in order to describe themselves, their work and their specialization. An example is shown in Fig. 1. In order to measure similarity according to this short description, we preprocessed corresponding fields of the two profiles. We removed the punctuation, lower-cased and tokenized the description, thereby creating two different token lists. The similarity of the two token lists is computed as the ratio of their common words, to the total number of all words in both description fields.

Location Measures. Our comparison utilizes the textual representation of the location field in a geospatial semantic way. We convert the locations provided in the different social networks to bounding boxes, with the use of the geonames ontology [12]. The similarity score of the two locations is defined by the following equation:

$$LocSim(l_1, l_2) = \begin{cases} Bbox(l_1) / Bbox(l_2) & \text{if } Bbox(l_1) \subseteq Bbox(l_2) \\ Bbox(l_2) / Bbox(l_1) & \text{if } Bbox(l_2) \subseteq Bbox(l_1) \\ 1 / (1 + \|l_1 - l_2\|_2 & \text{if } l_1,\ l_2 \in SC \\ 0.0 & otherwise \end{cases} \quad (3)$$

where Bbox represents the bounding box of the respective location and SC refers to the same country. The similarity score in all situations is normalized in the range [0.0 , 1.0].

For example lets assume that we have $SN_{location1}$ = “New York”, that appears in one social network and $SN_{location2}$ = “Manhattan”, that appears in the other. Since Manhattan is a borough of New York City, its bounding box will be included into the bounding box of New York city, as shown in Fig. 2. Thus, the similarity of the two locations is measured as the ratio between the covering area of Manhattan’s bounding and the area of New York City’s bounding box. Now suppose that we retrieve two locations that belong to the same country but their bounding boxes are not subsumed –$SN_{location3}$ = “Athens” and $SN_{location4}$ = “Sparta”. In this case, their similarity is computed by the Euclidean distance of the coordinates of the centres of two bounding boxes.

Figure 2: Example with bounding boxes in location measure.

Affiliation-Education measure. This measure attempts to match the current/ past affiliation and educational experience of the users, as stated in the social network profiles. In order to measure the similarity score we create two token sets, one for each corresponding network. Fig. 1 shows the profile fields that participate in this score. In LinkedIn's set we use the affiliation of current and past experiences and the educational schools, while in Twitter's set we use the userMentions (@ symbol in Twitter) that appear in the user's tweets. While neither of the two token sets include duplicates, the token set obtained from Twitter contains additionally the frequency of each userMention. An additional practical problem with userMentions is that they appear in an abbreviated form. So, there is a need for a textual comparison measure that is suitable for sub- string matching. Based on the related survey [4], the Smith-Waterman distance measure seems adequate, because it combines edit and affine gap distances. We measure the similarity between each pair of tokens in the two token sets and keep only those similarity scores that exceed a predefined threshold t. Then we weigh the resulting scores according to the frequency of a userMention in Twitter profile. Therefore, the overall similarity score is calculated as shown in the following equation:

$$\sum_{i=1}^{n}(score_i \times freq_i) \Big/ \sum_{i=1}^{n} freq_i \qquad (4)$$

where $score_i$ is the Smith-Waterman similarity score of a pair of tokens that is above the threshold t and $freq_i$ is the frequency of appearance of the specific userMention in the user's tweets. The weight indicates a significance estimate of the corresponding userMention.

Achievements measure. It is common that users highlight their professional achievements and their job specialization in the short biography field of their profile. The main idea is based on the observation where the words that a user often provides in description field in Twitter, belong to the same family with the ones that she provides for her job, publication etc in LinkedIn. We attempt to capture this by using SoftTFIDF metric, which takes into consideration "similar" and not only identical tokens [4]. We compose a textual summary of the most significant professional achievements of a user, as she provides in LinkedIn:

we combine current and past job experiences and the corresponding affiliations, professional specialization, projects and publications that she has participated in. The similarity between this "profession summary" and the short biography in Twitter is computed with the use of SoftTFIDF.

3.3. Classification

As mentioned above, the various similarity measures are used to built similarity- vectors. These vectors are then classified in order to achieve the required user identification. Below we describe the different classification approaches that we tested.

Baseline classification results. As a baseline we calculate the average of the scores in the similarity vectors:

$$AvgScore(V) = \sum_{i=1}^{n} \frac{(score_i)}{n} \quad (5)$$

where $score_i$ corresponds to the respective score in the similarity vector. The higher the score, the more likely it is that the corresponding profiles belong to the same user.

Binary Classifiers. A different way to classify similarity vectors is by training binary classifiers. For each profile set, we declare as "match" the profile that is assigned maximum probability by the classifier, depending on the classes' distribution of the respective binary classifier. The classifiers that we tested are:

- **Decision Tree:** In our study, we experiment with the C4.5 decision tree and use pruning to avoid overfitting.
- **Naive Bayes**
- **KNN:** In our study, we set the value of k to 5. Moreover, the nearest neighbors are determined by the Euclidean distance of the pair to the training instances.
- **NBTree:** This is a hybrid approach that involves Naive Bayes and Decision Trees classifiers.

- **DTNB:** This is a hybrid approach that involves Naive Bayes and Decision Tables classifiers.

4. Experimental Results

4.1. Data Collection and Experimental Setup

The collection of the data was based on name search, as denoted in Section 2. We started with a list of target users in mind, e.g. Katerina Zamani, where each one had a different name. Given the name of a particular target-user, we gathered the first 25 profile-results from each network, using the networks search engine. Thus, we created two sets of profiles (one for each network), each set containing the results of the search for a particular name. Specifically, each data set contains 262 profiles sets and we gathered 2766 LinkedIn profiles and 3373 Twitter profiles in total. The aim of our study was to identify within each such set only the profile of the target-user, given the users profile in the other network, e.g. given Zamani's profile in Twitter, we wanted to identify the profile of the same person in LinkedIn, among the set of profiles that the search for Katerina Zamani has returned. Each comparison produces a similarity vector, as described in Section 3.2, which is classified as a match or not. In each set we identified one profile as the correct match, while all others were considered mismatches. In our experiments we use two different datasets corresponding to the direction of the identification, i.e. starting with a profile from LinkedIn we compare it against the profiles of the corresponding set in the Twitter dataset and vice versa. Henceforth, we refer to the former task as Twitter identification and the latter as LinkedIn identification.

Missing Values. It is common that users do not complete every fill in all fields of their profile. This influences the performance of our approach because many profile fields that we use, are not available. Table 1 presents the percentage of missing fields for each similarity metric. The metrics with 0% of missing values denote that the respective fields are compulsory during the user's registration.

Table 1: Percentage of missing values

SN/metric	Name metric	Description metric	Location metric	Affiliation-Education metric	Achievements metric
LinkedIn	0%	67%	0%	17%	8%
Twitter	0%	42%	47%	22%	42%

Imbalanced Data. The nature of the identification problem across social networks results in considerable imbalance between the two classes (match vs. mismatch). In our study, only 9.5% of the LinkedIn profiles and 7.8% of the Twitter profiles comprise the minority (match) class. This imbalance can cause problems during training for some classifiers. In order to handle this issue, we suggest a procedure during the testing phase of classification.

4.2. Results for Separate Measures

In this section we evaluate separately each similarity measure that we used. Taking into consideration the large amount of missing values and how this could nfluence the accuracy of classification, we examined the following solutions:

- **Set a default score:** We set 0.5 as a default similarity score, when the score cannot be calculated. It was worth recalling that all scores are normalised in the range [0.0, 1.0].
- **Set the average score:** We set the missing similarity score to the average value of the similarity scores, that can be computed from the available fields. This average score is different for each metric and it depends on the measured similarity scores of the respective measure.
- **Set the median score:** The basic idea of this approach is similar to previous one, but instead of the average, we use the median value of the computed similarity scores.

In particular, we compute the recall of each similarity score separately. Note that precision is the same as recall here, since all methods are required to return

exactly 262 matches. Specifically, we select as the most likely matching set the one with the maximum similarity score, i.e. for each profile set we define as "match" the pair with the maximum average score. Table 2, provide the results for the two datasets (LinkedIn identification and Twitter identification), and for the different strategies to deal with missing values.

Table 2: Recall for LinkedIn and Twitter identification for different measures and different strategies for missing values. Results are presented as percentages to facilitate readability.

Iden/tion Type	Strategy	Name-Measure	Desc. Measure	Location Measure	Affiliation Education Measure	Achiev. Measure	Baseline Classifier
LinkedIn I.	Default	68.70%	60.31%	67.94%	80.15%	83.59%	86.26%
	Average	68.70%	64.12%	69.08%	79.77%	87.02%	86.64%
	Median	68.70%	63.74%	68.70%	79.77%	87.02%	83.59%
Twitter I.	Default	90.84%	80.92%	75.57%	75.19%	74.81%	74.81%
	Average	90.84%	85.50%	82.44%	75.19%	79.77%	88.55%
	Median	90.84%	85.50%	79.78%	74.43%	79.77%	85.11%

We can notice that the score in the name metric in Twitter's identification case is much higher, due to the different nature of network's search engines. In addition, the high success scores of the two last metrics in LinkedIn's identification case, indicate the importance of the professional fields in the identification. Regarding Table 2 we conclude that the average score approach predominates in missing values problem, so we choose this for the rest of our experiments.

4.3. Results of the Trained Classifiers

At this subsection we refer to our classification strategy and we present the results from the different classifiers we use. To estimate the performance of our classifiers we utilize the k-fold cross validation technique. Due to the structure of

our datasets, we split our sets to 7-folds in order to test the 14% of the database each time.

Due to imbalanced data problem, we specify as “match” the pair with the maximum probability. This probability, which is derived from the distribution of the positive class during training, denotes the likelihood membership of the instance in that class [11].

Table 3: LinkedIn and Twitter identification results for various classifiers

Iden/tion Type	Metrics	Decision Tree	Naive Bayes	KNN (k=5)	NBTree	DTNB
LinkedIn I.	Accuracy	97.87%	98.09%	98.40%	98.68%	98.96%
	Precision	89.58%	90.35%	92.66%	92.28%	94.98%
	Recall	88.96%	89.69%	91.99%	91.59%	94.27%
	F-measure	89.27%	90.02%	92.33%	91.93%	94.62%
Twitter I.	Accuracy	97.93%	97.89%	98.57%	98.52%	98.61%
	Precision	86.49%	86.10%	90.73%	91.12%	90.73%
	Recall	86.49%	86.10%	90.73%	91.12%	90.73%
	F-measure	86.49%	86.10%	90.73%	91.12%	90.73%

As we can notice from Table 3, our approach performs well for detecting matches and especially with the use of NBTree and DTNB classifier. Even the low proportion of groundtruth data, the results for precision and recall in match class are satisfactory, so we achieve a high score in accuracy. Taking into consideration the ROC curves in the Fig. 3, we can conclude that DTNB outperforms the other classifiers in both cases (LinkedIn identification and Twitter identification).

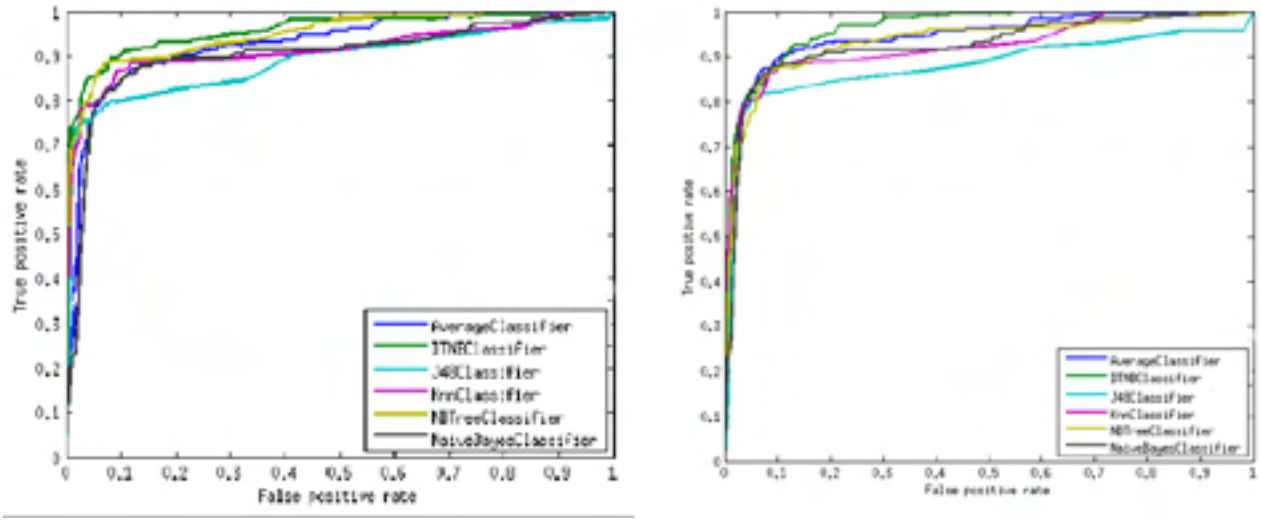


Figure 3: ROC curves of the classifiers. The left graph represents the LinkedIn identification task while the right graph the Twitter identification task

5. Conclusion and Future Work

In our work, we studied user identification in two popular social networks in order to support information verification. We used different similarity measures for different pieces of information provided by the user, and we combined them using supervised classification upon similarity vectors. As shown by our experiments, on the specific data set, using a hybrid classifier (DTNB) we can achieve a very high user identification performance.

A possible future extension of the presented work, would be the handling of class imbalance with a more sophisticated approach, either by using ensemble filtering (e.g SMOTE [6]), or by setting higher weights to the matches during training [1]. Moreover, we could enrich location information provided by the users with estimations of locations as mentioned by the user in tweets or job descriptions, as [5] suggests. Finally it would be interesting to study the potential contribution of our approach to the difficult problem of identifying fake or compromised account in social networks [3].

References

[1] Goga, O., Perito, D., Lei, H., Teixeira, R., Sommer, R.: Large-scale Correlation of Accounts Across Social Networks. Technical report (2013)

[2] Iofciu, T., Fankhauser, P., Abel, F., Bischoff, K.: Identifying Users Across Social Tagging Systems. In: L. A. Adamic, R. A. Baeza-Yates, S. Counts, ed. , ICWS. The AAAI Press (2011)

[3] Egele, M, et. al: COMPA: Detecting Compromised Accounts in Social Networks. NDSS (2013)

[4] Elmagarmid, A. K., Ipeirotis, P. G., Verykios, V. S.: Duplicate record detection: A survey. Knowledge and Data Engineering, IEEE Transactions on 19, 1–16 (2007)

[5] Chen, Y., Zhao, J., Hu, X., Zhang, X., Li, Z., Chua, T. S.: From Interest to Function: Location Estimation in Social Media. In: AAAI (2013)

[6] Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P.: SMOTE: synthetic minority over-sampling technique. J. Artificial Intelligence Research 16, 321–357 (2002)

[7] Moreau, E., Yvon, F., Capp, O.: Robust similarity measures for named entities matching. In: Proceedings of the 22nd International Conference on Computational Linguistics, vol. 1, pp. 593–600. Association for Computational Linguistics (2008)

[8] Cohen, W., Ravikumar, P., Fienberg, S.: A comparison of string metrics for matching names and records. In: Kdd Workshop on Data Cleaning and Object Consolidation, vol. 3, pp. 73–78 (2003)

[9] Malhotra, A., Totti, L., Meira Jr, W., Kumaraguru, P., Almeida, V.: Studying User Footprints in Different Online Social Networks. In: Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining, ASONAM, pp. 1065–1070. IEEE Computer Society (2012)

[10] Vosecky, J., Hong, D., Shen, V. Y.: User Identification across Multiple Social Networks. In: Networked Digital Technologies, First International Conference on NDT'09, pp. 360–365. IEEE (2009)

[11] Machine Learning Group at the University of Waikato http://www.cs.waikato.ac.nz/~ml/index.html

[12] GeoNames Ontology, http://www.geonames.org/

Development of Data Mining Tools for Identifying Structural Determinants that Dictate Protein-Ligand Interactions

Anaxagoras A. Fotopoulos (anfotopoulos@di.uoa.gr)
Athanasios V. Papathanasiou (thanospap@di.uoa.gr)

Abstract

Modelling binding sites of enzymes is a fundamental but rather demanding task, of increased complexity since the residues forming these sites are not rigid. Similarly, studies concerning binding of a ligand, at such a site and complex formation, raises difficulties mainly because most of the structural determinants that control binding are not known. Using a combination of sampling and statistical analysis, we contribute towards developing much more accurate binding affinity predictions for macro-molecular docking. To this end, we study benchmark protein families with known 3D structure with the aim to identify specific geometric parameters for modeling their binding cavities. We start by studying the boundaries within which every residue in those cavities can move, in 3D Euclidean or in conformational space. Key methods employed include structural alignment of secondary structure elements, RMSD heat-maps, sampling (e.g. in the space of rotamers), and (generously) allowed regions as defined in the Ramachandran plot. Our tools involve powerful methods, such as alpha-shapes, nearest-neighbor search and clustering, adapted to the specific context. The developed methods were tested on a subset of kinases proteins with known 3D structure, which offer a number of target sites for one or several ligands. Sets of rotamers were produced by sampling the chi angles and testing steric clashes, then clustered in a 2-level hierarchical process. For each cluster, representative polyhedral shapes were produced which can thereafter be exploited for ligand screening.

Keywords: Conformers, Hierarchical Clustering, Structural Determinants, Conformation Space, Data Mining

Advisors

Ioannis Emiris, Professor, Evangelia Chrysina, Research Officer NHRF-IBMC

1. Introduction

During the last decades a number of drugs have been launched to the market. However there is lack of specificity, which can cause various side effects. This has been the driving force for further research on the design of drugs that target only their receptors. More specifically the modelling of the binding sites of macromolecules is a rather demanding task, due to their lack of rigidity. The discovery of new methods and tools is expected to offer new opportunities. The current thesis focuses on the finding of structural determinants that dictate binding, which falls within the scope of the Structure-Based Drug Design approach, by detailed mapping of the cavity geometry. A novel rationale has been developed for the simulation of the movement of the catalytic site residues, using data analytics and geometric features. In this manner, an anticipation towards developing much more accurate affinity predictions for complex formation and macro-molecular docking is achieved [1]. Emphasis was given on the analysis of the different conformations in the active site. The manifold degrees of freedom and the restraints imposed by the backbone increased the overall complexity. These issues are addressed in the presented methods. The production of different conformers is based on the simulation of chi angles rotations coupled with application of filters to take into account steric clashes. Conformers are clustered based on a two level hierarchical analysis. A biological validation with keyword matching techniques of the produced clusters is also available. Except from the typical clustering of protein structures with the use for superposition and RMSD distance, a new innovative method is presented: the multi-dimensional k-means clustering without superposing protein structures. After the clustering of conformers, representative polyhedral shapes were produced which defined the local minima and maxima of the XYZ co-ordinates and can be further exploited in either rational or random ligand screening approaches. Furthermore, the shapes may also serve as a template that would reveal the

complementary shape of a potential ligand.

To further aid researches in protein analysis, a series of bioinformatics tools were developed with functions such as the extension of the currently available statistical visualization tools, the data mining of present information from online databases, the optimized simulation for a faster execution with parallel programming methods and the expansion of the active site for surrounding amino-acids with geometrical and secondary structure considerations.

The drug design efforts based upon the 3-dimensional structure of a macromolecular target is considered to be a hallmark of modern molecular design strategies. The structure-based drug design has already made a significant impact in the drug discovery process with more than 35 newly approved drugs l aunched in the market. In the post-genomic era, many important drug targets are emerging and the structure-based design is expected to offer even more new opportunities. Despite the market dynamics, structure-based drug design has not reached its full potential and the newly introduced methods in this area of research will play a vital role in the drug discovery endeavors.

2. A novel Rationale for the Computation of Conformers with Clustering Techniques

2.1. Protein Dataset

New types of experiments, such as the “Human Genome Project”, completed in 2003, produced large amounts of data. Various online databases where developed throughout the years including GenBank [2] and Protein Data Bank [3].

Kinase inhibitors are considered of high importance, as important drug targets for the development of anti-cancer therapies and much effort has been made towards the structural analysis of the binding sites of various protein kinases [1]; they are also involved in a plethora of metabolic pathways. Several studies have been presented over the years and as it has been reported that the use of a single kinase structure in docking studies may produce false negative results. In order to overcome this problem protein multiple structures were taken into consideration for the analysis; in order to have an adequate number of representative structures. Thus a dataset of 100 kinases from diverse families was obtained

from Protein Data Bank [3]. All kinases were related with at least one journal reference in PubMed.

2.2. Distance Matric Calculation

The comparison of 3D structures is an important task [4] used in finding the structural evolution of proteins or protein domains [5]. Two different approaches have been followed in the computation of the distance matrix of the protein set: a) Computation with Matlab commands, and b) Computation the use of RCSB PDB Comparison tool [6] with alignment provided from jFATCAT rigid algorithm [7]. Both methods use the Root Mean Square Distance (RMSD), which is calculated between equivalent atoms in two structures and is the most common distance metric for the expression and analysis of structural similarity.

2.3. Protein Clustering with Hierarchical Analysis

After the calculation of all pairwise differences of the kinase protein structures, a hierarchical clustering algorithm will be performed for the production of the dendrogram that is shown below. From the original clustering, the user can define the number of clusters based on an input or dynamically by defining the cutoff of leaf distance from the bottom to the top of the hierarchy. In the presented example (Fig. 1) the clusters are defined from a cutoff of the 60% of the tree height. Each cluster is represented with a different color. The role of hierarchical clustering is vital in the present thesis, as the divide and conquer computational approach has been followed in conformer analysis. The main aim of clustering's usage is the overall reduction of the computational cost and the limitation of the analysis to structural homologues.

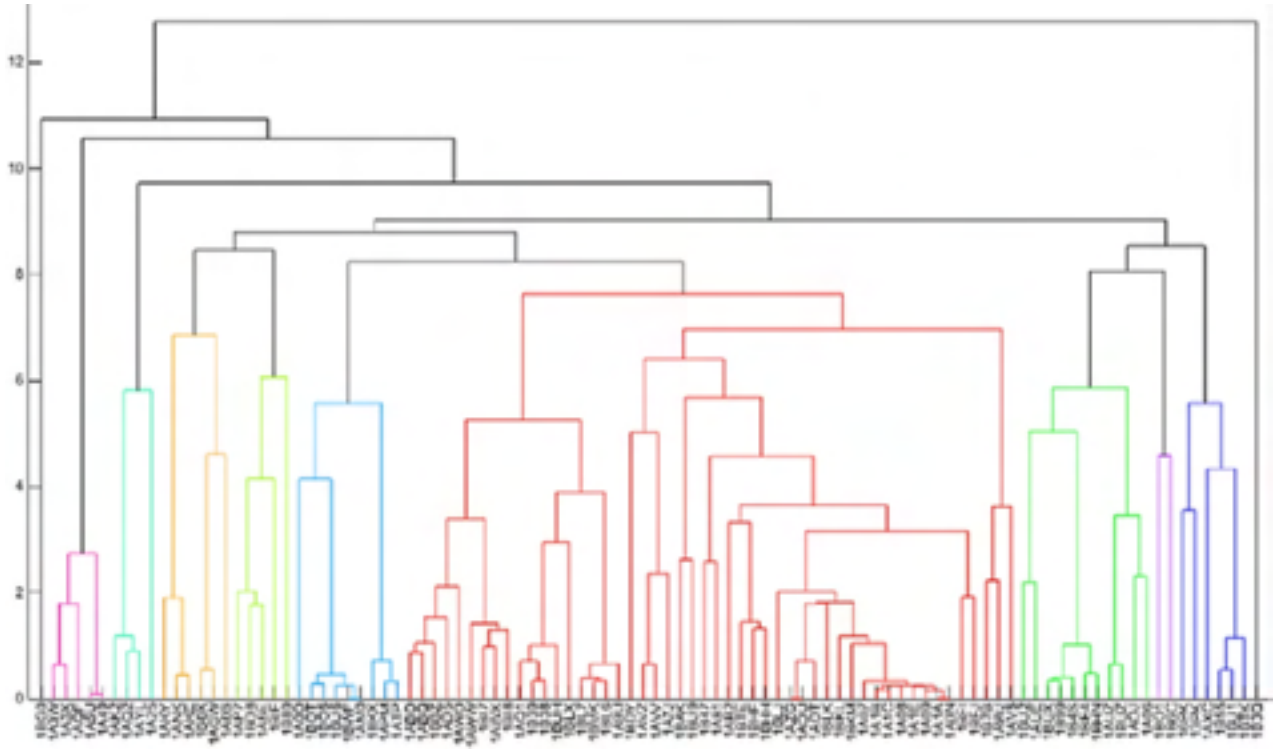


FigURE 1: Hierarchical Clustering of 100 kinases. Clusters are produced from a cutoff to the 60% of the maximum leaf height.

2.4. Protein Clustering with Multidimensional k-Means

In this section a new novel method is introduced for the clustering of proteins sets. The common process is the superposition of each protein with each other and then the computation of their RMSD distance. Moreover various problems have been observed in the superposition of different size proteins. Thus, an effort has been made towards overcoming this problem. More specifically, a process has been made for the reduction of the overall computational cost by increasing the problems' dimensions.

The methodology that was conducted is the following: a) For a list of 100 kinases the Ca trace is computed and the XYZ co-ordinates are kept in a vector of the following form $\{X_1, X_2, \ldots, X_n, Y_1, Y_2, \ldots, Y_n, Z_1, Z_2, \ldots, Z_n\}$. b) After the finish of the process, due to the fact that all vectors should have the same length, the maximum length of atoms is found and all other vectors are filled with zeros until they all have the same length. c) A multidimensional k-means

algorithm is implemented. d) For each of the k clusters a hierarchical cluster is performed based on the distance of each clustered entity to the k-th cluster. Hierarchical clustering is used in each cluster for the production of local dendrogram and cannot be implemented to the whole set, due to the fact that the distance of each entity is computed for the cluster's centroid and there aren't distances between proteins. Thus the disadvantage of this method the inability to cluster the clusters to a hierarchy. However it important to denote, that the implemented process does not use superposition and doesn't compute all pairwise distance, instead a reduction in complexity has been made by downsizing a problem of distance computation from $all\text{-}vs\text{-}all$ to $all\text{-}vs\text{-}k$, where k is the number of the clusters.

2.5. Clustering Evaluation

In computational biology it is vital to validate results in order to evaluate the success rate of an in silico simulation. Moreover, the discovery of biological notion in the provided results is of high interest for the explanation of possible protein correlations. In this chapter a novel approach has been developed for the "biological validation" of protein clustering results with text mining techniques. With the term "biological validation" an answer is given to the question of the "How closely related are two proteins from biological aspect?". The initial approach was to find correlations in the PDB file's description; however with this technique higher complexity and "noise" existed. Thus a simpler and more accurate approach was followed. More specifically, as it can be easily underlined from the following table, certain textural similarities exist among the protein kinase set according to their keywords. Hence, for each cluster of the previous subchapter, all possible pairwise combinations between proteins were undertaken and for each pair 1-vs-all comparison was made for each keyword of the one protein with the other. If the keyword were the same, then at the corresponding position of the pair, in the scoring matrix of the specific cluster the score was increased by one. In this manner, in a protein set (a provided cluster) if two proteins have at least one corresponding textural correlation (two keywords were the same), then their clustering in the same group was made right. In the keywords of a PDB id, the protein family will be referred in most of the cases. Thus the possible correlation of keyword matching, will refer probably to the same protein family notion. In order to provide even better

results the keyword “kinase” was removed in a pre-processing operation.

For the dendrogram mentioned previously, the following table shows for each cluster the number of proteins that didn’t had a match with any keywords of the other proteins of the specific cluster, along with the number of “proper biological clustering”. The last term means for the case of the first cluster that in 9 protein structures, the 1 was found “irrelevant” from biological aspect, thus the 8 proteins or 88.99% were biologically clustered in a proper manner.

Table 1: Validation results for multidimensional k-means

Cluster Number	Number of proteins in the cluster	Number of non-relevant protein structures	Percentage of proper biological clustering
1	9	1	88.89
2	50	0	100
3	5	2	60
4	6	1	83.33
5	6	0	100
6	13	0	100
7	4	1	75
8	5	0	100
9	1	1	0
10	1	1	0

2.6. Selection of Catalytic Site Vicinity

The detailed information of the catalytic residues of the enzyme active site and the residues in the vicinity are essential in understanding the relationships of protein structure and its functions [8].

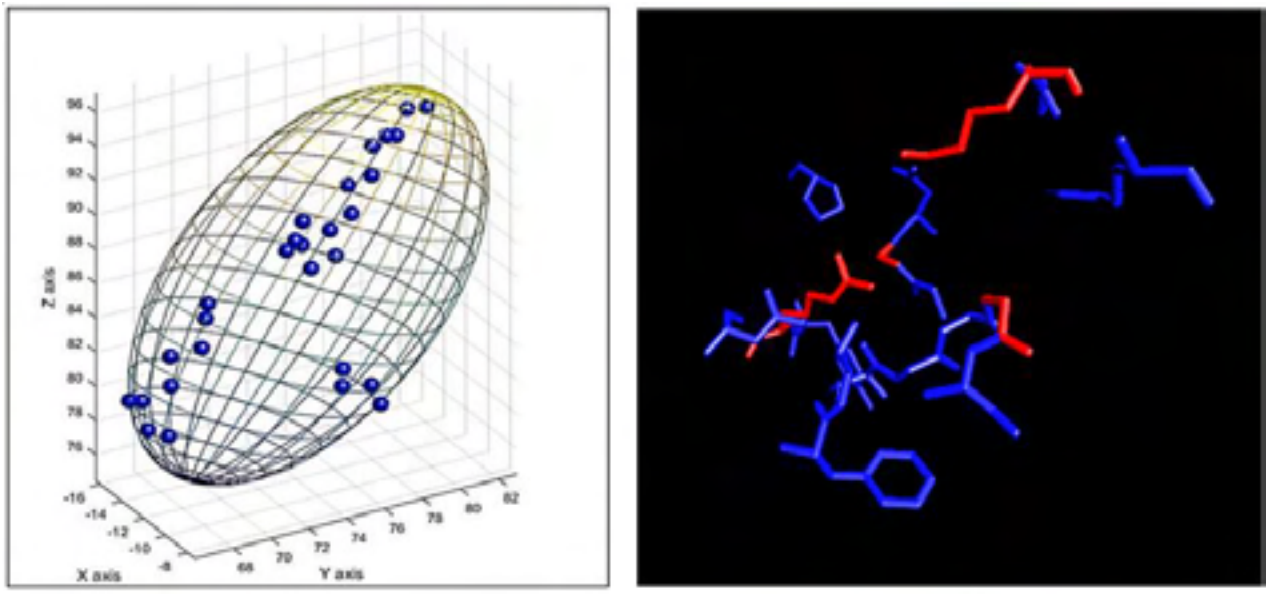
The CSA (Catalytic Site Atlas) provide curated annotations for 968 protein entries that have been deposited with the PDB [3] and uses a sophisticated method for transferring the annotations the homologous structures increasing the robustness of annotation transfer. Moreover the curated entries are used along with sequence comparison for the generation of 3D templates of the catalytic sites,

which can thereafter find catalytic sites in new structures [8].

The develop program, initially receives a user-defined PDB id input from a simple GUI. Subsequently the possible catalytic site is extracted with the use of text mining techniques and regular expressions. The available sites are therein presented in a menu of choices, where the user selects based on his/her preference. In this manner, the conformation analysis that will be described in next steps can take place in any given residues of a protein structure.

In many studies, the surrounding residues of the active site, may affect the inhibitor, even from distal position in an unclear manner. Mutations may hurt the conformational equilibrium, which could result in weaker binding [9]. Thus it is important to take into consideration the surrounding region of the catalytic site. According to literature, various techniques haven been used in pocket recognition and could also be used to the expansion of the catalytic site including grid based approaches, sphere coating approaches and alpha & beta shapes [10].

Four approaches have been developed for the identification of the active site's neighboring residues: a) User defined sphere b) Minimum volume enclosing sphere c) Minimum volume enclosing ellipsoid and d) Inflated Alpha shapes.



(a) (b)

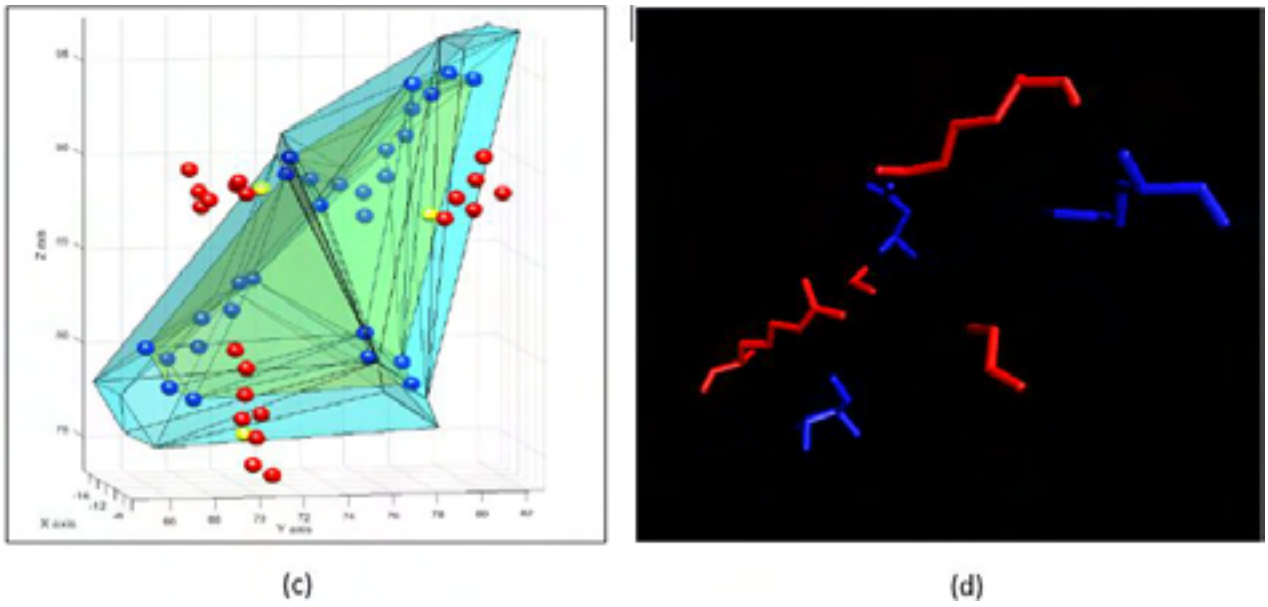


**Figure 2: (a, b). Minimum volume enclosing ellipsoid. The blue atoms suggest the residues of the catalytic site.
(c, d). In green the original alpha shape is shown. The blue spheres represent the atoms of the catalytic site residues.
After the inflation of the alpha shape by 30% which is shown in turquoise the yellow atoms of the surrounding residues are part of the new shape. Thus the catalytic site is extended by three new residues.**

For the choice of the user defined sphere a simple GUI asks for the sphere radius. After the definition of the radius, the program finds the centroid of the active site's atoms and draws a sphere. Thereafter, all atoms of the protein structures are checked on whether they are part of the geometric shape. For those atoms that are part of the shape its corresponding amino acid is added to a list of neighboring residues. The other option is to define the minimum volume enclosing sphere. With this method the sphere encloses the minimum space that can have so the total of all points (of the active site residues) can be enclosed. Another option is the minimum enclosing ellipsoid. This method is operated in exact the same way with the minimum enclosing sphere with the difference that the shape is an ellipsoid. A classical reference to the definition of ellipsoid is the given in [11]. For the computation of minimum volume enclosing ellipsoid the open source program MinVolEllipse has been used [12].

The main difference between the minimum volume enclosing ellipsoid and

enclosing sphere is that due to topological particularities the ellipsoid may reduce the "noise" of non-relevant residues that could be selected from the sphere. Thus, the ellipsoid will reduce the number of selected residues and increase the accuracy (Fig. 2a, Fig. 2b).

The last option is the selection of surrounding residues based on alpha shapes. This method is even more subtractive in the selection of other residues that are not in the active site, this occurs due to the fact that the rolling ball will minimize the available space between the atoms of the catalytic site residues that conform triangles. Thus the selection of the rolling ball radius is vital for a proper use of this method. The radius should not be relatively small, as it will create separate shapes for the residues.

In this method, to ensure that all atoms are part of the same shape the maximum of all pairwise atom distances was selected as the radius of the rolling ball. In sequel an outwardly inflation of the polygonal shape is performed. As it will be shown in the following picture the inflation of the alpha shape of the catalytic site encapsulated new atoms from surrounding residues (Fig. 2c, Fig. 2d). The inflation was made by scaling the original coordinates of the atoms of the catalytic site by 30% and then from XYZ coordinates, the centroid difference (of the original and the inflated point-clouds) was deducted. In this manner, a shifting to the original centroid for the inflated point-cloud was made. From the inflated point-cloud the new alpha shape was computed and shown along with the original one (Fig. 2c). Hence it can be concluded that this method provide even more specificity in the expansion of the catalytic site in an automated manner. This method is vital in the reduction of the available degrees of freedom for the production of the conformers.

2.7. Probabilistic Computation of Rotamers and Fast Computation of Non-Steric Conformers

During the structural analysis of a protein it is important to focus on the active site and its topological & geometric characteristics. In this section particular attention was given to the production of different conformers at the catalytic site. The different side-chain conformations are produced based on a probabilistic selection through the online rotamer library of Dynameomics [13]. At a glance, rotamer libraries show the probability distribution of an observing

residue for a given rotamer. According to the literature, there are two major libraries a side chain only library from Richardson laboratory [14] and a library that includes main chain conformations of Dunbrack laboratory [15]. As stated in [13] “But these techniques are not without their own problems. The crystal structures themselves, and especially the filtering techniques employed, may yield an overly static view of protein structure. Flexible proteins that crystallize at lower resolution or inherently flexible amino acid conformations with high B-factors are excluded. The libraries use crystal structures instead of solution structures and may suffer from artifacts such as crystal contacts, effects of crystallization conditions, or changes from mutations or truncations necessary to improve crystallization quality. In addition, the number of structures determined under cryogenic conditions is increasing, which can also skew the distributions”. Dynameomics library simulate the native state and unfolding behavior of representatives of all autonomous protein folds. More particularly 807 proteins were analyzed, totaling 86.217 residues with at least 31.000 samples of each residue (2.7x109 rotamers). Thus for the above reasons Dynameomics library was selected as for our in silico simulations.

Initially the program accepts as an input a PDB id and downloads its structure from PDB [3]. Then the user is asked to give a probability cutoff for the extraction of the chi angles from the site of Dynameomics [14]. From the available chi angles only those that have a probability above the selected cutoff, will be selected. The chi angles for all aminoacids are downloaded and can be used of active site expansion. After the selection of residues of the active site, user is asked on whether to extend or not the active site. After all the previous stages, the process of the different conformers is initiated. At first, all possible combinations from the probable rotamers between aminoacids are computed. Thus after the produced list of the determined chi angles for the specific residues of the (extended) catalytic site, for each combination every amino acid is taken and the chi 1 to chi 5 (chi angle number depends on the amino acid type) angles are sequentially changed by the difference of the rotamer probability matrix with the specific chi angle. Moreover, after the end of a combination of chi angles, a steric clash identification process is initiated.

As it can be easily derived from the aforementioned pipeline, the conformers simulation is a computationally demanding process. For example for the case of 13PK protein structure by extending the catalytic site with the minimum vol-

ume enclosing ellipsoid method, 13 amino acids would have been selected for analysis and subsequently 150 billion different combinations would be needed for a complete computation. This, would be unfeasible from points of memory and computational time. The computational cost for distant amino acids could be avoided through a divided and conquer approach, where the computational cost could be minimized via a dimensionality reduction. Initially, a hierarchical clustering has been performed between amino acids to identify those lying close in the 3D space. For the lower clusters of the hierarchy, all possible combinations of the conformers were calculated. Hence, consecutive merging of all sub-clusters, was performed as moving from the bottom to the top of the hierarchy, until only one group exists. The process described was further accelerated by employing multithreading techniques. All non-steric conformers are saved in PDB file format for a further analysis or visualization through other programs.

2.8. Conformers Clustering with Iterative Closest Point Alignment

Previously, the process of the calculation of conformers was shown. As mentioned the conformers were exported as PDB files. In this chapter an effort will be made for the clustering of conformers and the analysis that derive. Initially from the previous-referred hierarchical clustering of the 100 kinases 10 clusters have been produced as it was shown. For the cluster of the following image all different conformers were produced totaling 107,016 in number approximately 20,000 for each protein. Several issues have reported over the years with the typical use of RMSD from superposition (rotation, translation) due to the fact the number of atoms may vary [16]. Thus the Iterative Closest Point (ICP) technique has been used the alignment of the point clouds of the conformers. As referred to [17] “ICP aligns and registers an unlabeled set of point p to a model set X by iteratively alternating between registration and alignment steps. Registration is obtained by finding the closest point $y \in X$ to each point $p_i \in p$, resulting in the corresponding set Y. The point clouds are aligned by finding the optimal rotation matrix and translation vector that superposes p onto Y. The steps are repeated until the change in mean square error between p and Y falls beneath a desired threshold”. ICP have been also used instead of the DALI alignment in [18]. After the alignment the RMSD has been used for the calculation. It should noted here that the divide and conquer approach is also present during this methodology. The initial cluster of 100 kinases provide the ability to

make a 2-level clustering. In this manner, the clustering of the conformers is simplified and more problems regarding the different size of atoms are usually overcome due to the fact that the clustered proteins are rather structural similar. In other approaches the energy calculation is used for the selection of conformers. In the presented method, this issue is addressed in a more statistical way through the computation of the exemplars of the clusters. Assuming that 10% of the conformations would be selected as the cluster exemplars, the maximum number of leafs is set to this metric. Each leaf has an index matrix, which contains all corresponding conformers ids. Thus for each leaf the median value of a local distance matrix is selected as exemplar.

2.9. Visualization of Structural Boundaries

For a larger dataset of kinases and consequently conformers it is considered important to understand the results of the hierarchical clustering from a structural point of view. Towards this, alpha shapes have been used for the structural representation of each cluster. In this manner an overview can be given regarding the movements of the residues. The 3D polygonal shapes shown in Fig. 3, can been further exploited in either rational or random ligand screening approaches. In additional, it would also serve as a template that would reveal the complementary shape of a potential ligand. More specifically, the above alpha hulls represent the local minima and maxima of the XYZ co-ordinates for the selected group of conformers. With the knowledge of such information, ligands may be designed in a more efficient manner by taking into consideration the new-derived topological restraints. In practice, if the alpha shapes have a channel - like shape, ligands of circular shape will not bind properly. The derived ligand scaffolds could act as stable structural motifs that could contribute towards the development of much more accurate binding affinity predictions for macro-molecular docking. The conformational stabilization of the catalytic site from the interior which residues could adopt would favor ligand binding with structurally compatible ligands, while leaving most of the exterior amino acid side chains accessible to solvent. Hence, a proper modeling of binding cavity may utilize even more functional protein structures.

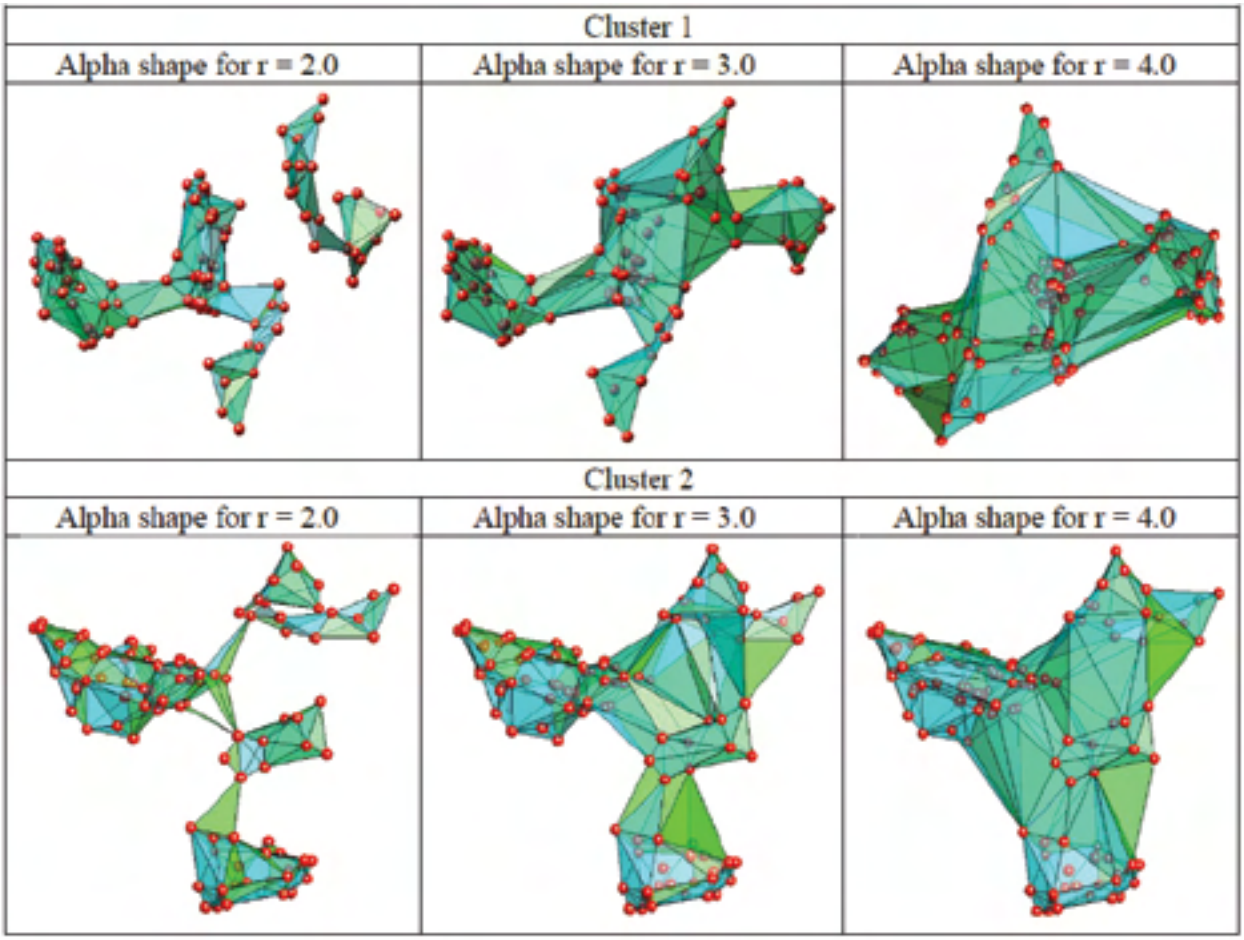


Figure 3: Representative alpha shapes for 2 clusters of the hierarchical analysis of conformers. For rolling ball of radius = 2.0 Å, 3.0Å and 4.0Å.

3. Conclusions

The different geometrical methods that were used to expand the catalytic site provided a better understanding of the surrounding region and the identification of the cavity. However the increased degrees of freedom of the side-chain angles for an adequate number of residues (>10) is still a computational demanding process. Our divide and conquer approach in the conformers' simulation reduced dramatically the overall execution time and could reinforce the in silico experimentations. The same methodology of simplification was implemented in the 2-level hierarchical clustering of conformers. Initially 2 methods of clustering whole proteins were used: a) classic superposition of proteins for pairwise RMSD distance, then hierarchical clustering, and b) multidimensional k-means for

partial clustering that ignores pairwise distances and superposition. Both methods were validated with a keyword matching method and provided acceptable results, namely a large number of proteins in each cluster are of the same family. Method (b) provided a better distribution of the clusters and a faster execution and seemed promising. For a cluster, different conformers were constructed and then clustered. It was found that conformers of close structural similarity are entangled in a similar way with their parent - proteins. Thereafter for each cluster of conformers representative polyhedral shapes were produced and can outline the local minima and maxima of the XYZ coordinates and could be exploited in ligand screening or for the design of the complementary shape of a potential ligand. These structural determinants may contribute in more accurate affinity predictions for docking. However the proper selection of the rolling ball radius in the alpha - shapes method that was used still remains unclear.

References

[1] A. Fotopoulos, A. Papathanasiou, E. Chrysina, and I. Emiris, A novel rationale for computation of potential conformers through clustering, Proceedings of the 10th conference of the Hellenic Society for Computational Biology & Bioinformatics HSCBB15, Athens, Greece, 2014.

[2] D. Benson, M. Cavanaugh, K. Clark, I. Karsch-Mizrachi, D. Lipman, J. Ostell and E. Sayers, “GenBank,” Nucleic Acids Res., vol. 41, pp. D36-42, 2013.

[3] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov and P. Bourne, “The Protein Data Bank,” Nucleic Acids Research, vol. 28, pp. 235-242, 2000.

[4] O. Carugo and F. Eisenhaber, “Probabilistic evaluation of similarity between pairs of three-dimensional protein structures utilizing temperature factors,” J. Appl. Cryst., vol. 30, pp. 547-549, 1997.

[5] F. Domingues, W. Koppensteiner and M. Sippl, “The role of protein structure in genomics,” FEBS Lett., vol. 476, pp. 98-102, 2000.

[6] A. Prlic, S. Bliven, P. Rose, W. Bluhm, C. Bizon, A. Godzik and P. Bourne, “Pre-calculated protein structure alignments at the RCSB PDB website,” Bioinformatics,

no. 26, pp. 2983-2985, 2010.

[7] Y. Ye and A. Godzik, “Flexible structure alignment by chaining aligned fragment pairs allowing twists,” Bioinformatics, vol. ii, no. 2, pp. 246-255, 2003.

[8] N. Furnham, G. Holliday, T. A. De Beer, J. Jacobsen, W. Pearson and J. Thornton, “Catalytic Site Atlas 2.0: cataloging catalytic sites and residues identified in enzymes,” Nucleic Acids Research, 2013.

[9] N. Goldfarb, M. Ohanessian, S. Biswas, D. J. McGee, B. Mahon, D. Ostrov, J. Garcia, Y. Tang, R. McKenna, A. Roitberg and B. Dunn, “Defective Hydrophobic Sliding Mechanism and Active Site Expansion in HIV 1 Protease Drug Resistant Variant Gly48Thr/Leu89Met: Mechanisms for the Loss of Saquinavir Binding Potency,” Biochemistry, vol. 54, pp. 422-433, 2014.

[10] J. Kim, C. Won, J. Cha, K. Lee and D. Kim, “Optimal Ligand Descriptor for Pocket Recognition Based on the Beta-Shape,” PLoS ONE, vol. 10, no. 4, 2015.

[11] H. Goldstein, Classical Mechanics, 2nd ed., 1980.

[12] N. Moshtagh, “Minimum Volume Enclosing Ellipsoid,” 2006. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/9542-minimum-volume-enclosing-ellipsoid. [Accessed 16 02 2015].

[13] A. Scouras and V. Daggett, “The dynameomics rotamer library: Amino acid side chain conformations and dynamics from comprehensive molecular dynamics simulations in water,” Protein Science, vol. 20, pp. 341-352, 2011.

[14] S. Lovell, J. Word, J. Richardson and D. Richardson, “The penultimate rotamer library,” Proteins, vol. 40, pp. 389-408, 2000.

[15] R. J. Dunbrack and F. Cohen, “Bayesian statistical analysis of protein side-chain rotamer preferences,” Protein Sci., vol. 6, pp. 1661-1681, 1997.

[16] J. Phillips, L. Ran and C. Tomasi, “Outlier Robust ICP for MInimizing Fractinal RMSD,” in Sixth International Conference on 3D Digital Imaging and Modeling, 2007.

[17] P. Markstein, “Computational systems bioinformatics,” in CSB conferece proceedings, Stanford, 2008.

[18] D. Xu, L. Hua and G. Tongjun, “Protein Structure Superposition by Curve Moment Invariants and Iterative Closest Point,” in 1st International Conference on Bioinformatics and Biomedical Engineering IEEE, 2008.

Τρισδιάστατες Κατασκευές με Χρήση Προκαθορισμένων Δομικών Στοιχείων

Γεώργιος Α Χρυσίνας (grad1133@di.uoa.gr)

Περίληψη

Σκοπός αυτής της εργασίας είναι η δημιουργία ενός προγράμματος, το οποίο θα τοποθετεί εικονικά τούβλα πάνω στο περίβλημα ενός τρισδιάστατου μοντέλου.

Τα βασικά κριτήρια για την τοποθέτηση των τούβλων είναι το να είναι εφικτή η πραγματοποίηση της κατασκευής στον πραγματικό κόσμο και η μεγιστοποίηση της σταθερότητας της.

Στη συνέχεια χρησιμοποιώντας τις πληροφορίες που έχουμε για τα τούβλα (θέσεις και ενώσεις), κάνουμε μία προσομοίωση φυσικής της κατασκευής. Έτσι μπορούμε να δούμε προσεγγιστικά την συμπεριφορά που θα είχε η κατασκευή αν κατασκευαζόταν στον πραγματικό κόσμο και κατά πόσον είναι στέρεη.

Λέξεις κλειδιά: Τρισδιάστατο Μοντέλο, Εικονική Κατασκευή, Δομικά Στοιχεία, Προσομοίωση Φυσικής, Τοιχοποιία

Επιβλέποντες

Θεοχάρης Θεοχάρης, Καθηγητής, Γεώργιος Παπαϊωάννου, Επίκουρος Καθηγητής, Οικονομικό Πανεπιστήμιο Αθηνών

1. Εισαγωγή

Υπάρχουν διάφοροι λόγοι για τους οποίους μπορεί κανείς να χρειάζεται ένα πρόγραμμα που τοποθετεί εικονικά τούβλα σε ένα τρισδιάστατο μοντέλο. Κάποιοι λόγοι θα μπορούσαν να είναι:

- Δημιουργία υφής τούβλων για το χρωματισμό του μοντέλου ενός κτιρίου.
- Προσομοίωση κατασκευών από τούβλα για μελέτη της σταθερότητάς τους.
- Υπολογισμός των θέσεων των τούβλων για το χτίσιμο τους από κάποια ρομποτική διάταξη.

Στην κάθε περίπτωση, έχουμε διαφορετικές απαιτήσεις από το πρόγραμμα. Για παράδειγμα στην περίπτωση της υφής δεν μας ενδιαφέρει ο εσωτερικός όγκος ενός τοίχου, αλλά μόνο το σχέδιο που προκύπτει στην επιφάνεια, αλλά αυτό δεν ισχύει στις δύο άλλες περιπτώσεις. Στην δεύτερη περίπτωση δεν ψάχνουμε τη βέλτιστη τοποθέτηση των τούβλων, αλλά προσπαθούμε να προσομοιώσουμε τον τρόπο χτισίματος ενός ανθρώπου. Επίσης θα μπορούσαμε να δεχθούμε μικρές παρατυπίες στις διαστάσεις των τούβλων.

Η τρίτη περίπτωση είναι αυτή στην οποία πλησιάζει περισσότερο και η υλοποίηση αυτής της εργασίας. Εδώ οι προδιαγραφές μπορούμε να πούμε ότι είναι οι εξής:

- Γίνεται χρήση τούβλων προκαθορισμένων διαστάσεων και ενός αριθμού στρογγυλών υποδιαιρέσεων αυτών.
- Αυστηρή τήρηση των διαστάσεων των τούβλων. Στον πραγματικό κόσμο τα τούβλα κατασκευάζονται με τυποποιημένες διαστάσεις οπότε προφανώς δεν πρέπει να αποκλίνουμε απ’ αυτές.
- Απαγορεύονται οι επικαλύψεις μεταξύ των τούβλων. Αυτό είναι προφανές ότι δεν συμβαίνει ποτέ στον πραγματικό κόσμο. Όταν όμως τα τούβλα αναπαριστώνται απλά από αριθμούς στη μνήμη του υπολογιστή, χρειάζονται πολλοί έλεγχοι για να βεβαιωθούμε ότι δεν υπάρχουν επικαλύψεις μεταξύ των όγκων τους. Αλλιώς όταν έρθει η ώρα της τοποθέτησης των αληθινών τούβλων θα διαπιστώνουμε ότι δεν υπάρχει διαθέσιμος χώρος.
- Η τοποθέτηση των τούβλων γίνεται με τέτοιο τρόπο ώστε να έχουμε καλή

πλέξη μεταξύ τους. Αυτό εξασφαλίζει τη σταθερότητα ενός τοίχου.

Το πρώτο σκέλος αυτής της εργασίας λαμβάνει υπόψιν του τα παραπάνω. Η υλοποίηση έχει γίνει σε C++. Η είσοδος είναι ένα μοντέλο το οποίο έχει σχεδιαστεί σε κάποιο πρόγραμμα τρισδιάστατης σχεδίασης (AutoCAD, 3D S tudio, Blender3D κλπ). Πάνω στο περίβλημα αυτού του μοντέλου τοποθετούμε τα τούβλα. Βασική προϋπόθεση για να είναι στέρεη η τελική κατασκευή είναι και η σωστή σχεδίαση του αρχικού μοντέλου. Αν για παράδειγμα το μοντέλο περιέχει τοίχους με μεγάλη κλίση, το πρόγραμμα αυτής της εργασίας θα τοποθετήσει και εκεί τούβλα, χωρίς όμως να δίνει οποιαδήποτε εγγύηση για τη σταθερότητα του αποτελέσματος. Δεν γίνονται δηλαδή τροποποιήσεις το αρχικό σχήμα ώστε να εξασφαλισθεί επιπλέον σταθερότητα. Με αυτό το θέμα ασχολείται μία άλλη εργασία, η [1].

Για να αυξηθεί η σταθερότητα της κατασκευής πρέπει τα τούβλα να είναι όσο το δυνατόν καλύτερα πλεγμένα μεταξύ τους. Θέλουμε δηλαδή τα άκρα ενός τούβλου να βρίσκονται περίπου πάνω από το μέσο των τούβλων του από κάτω επιπέδου. Αυτός είναι ο τρόπος χτισίματος που έρχεται αυθόρμητα στο μυαλό όλων. Στη πραγματικότητα υπάρχουν πολλοί άλλοι τρόποι χτισίματος που η εφαρμογή τους είναι κατάλληλη ανάλογα με την περίπτωση [4]. Ο τρόπος χτισίματος που αναφέραμε είναι γνωστός με τον όρο «δρομική τοιχοποιία» (stretcher bond). Αυτός είναι ο πιο απλός τρόπος χτισίματος, και έχει την πιο γενικευμένη χρήση. Το βασικό σκεπτικό του αλγορίθμου που χρησιμοποιούμε για να φτάσουμε σε αυτό το αποτέλεσμα, είναι το εξής. Θεωρούμε τις ενώσεις μεταξύ των τούβλων ως τα αδύναμα σημεία ενός τοίχου και φροντίζουμε σε διαδοχικά επίπεδα να είναι όσο πιο απομακρυσμένα γίνεται μεταξύ τους.

Είτε ένα τοίχος είναι επίπεδος, είτε ελαφρώς καμπυλωμένους, χρησιμοποιούμε το ίδιο σκεπτικό για την τοποθέτηση των τούβλων. Απλά σε έναν καμπύλο τοίχο τα τούβλα δεν εφάπτονται απόλυτα αλλά αφήνουν κάποια κενά ώστε να ακολουθήσουν την καμπυλότητα. Ένας τοίχος ορίζεται από διαδοχικά ευθύγραμμα τμήματα, τα οποία αν έχουν μικρές διαφορές στον προσανατολισμό τους μπορούν να δημιουργήσουν προσεγγιστικά καμπύλους τοίχους. Όμως αν η γωνία μεταξύ δύο διαδοχικών ευθυγράμμων τμημάτων, είναι μεγαλύτερη από κάποιο όριο που έχουμε επιλέξει, τότε εκεί έχουμε μία γωνία στον τοίχο. Στις γωνίες τα τούβλα πρέπει να τοποθετηθούν με διαφορετικό τρόπο. Σε έναν απλό τοίχο οι μικρές πλευρές των τούβλων ακουμπάνε μεταξύ τους. Στις γωνίες των κτιρίων όμως, έχουμε τις μικρές πλευρές των τούβλων να εφάπτονται

με τις μεγάλες. Η τοποθέτηση των τούβλων πρέπει να ξεκινάει από τις γωνίες και απαιτούνται ειδικοί υπολογισμοί για την σωστή τοποθέτηση. Παρόμοιο σκεπτικό ισχύει και στις διασταυρώσεις τοίχων.

2. Σχετικές Εργασίες

- Η εργασία [1] ασχολείται με την βελτιστοποίηση των γεωμετρικών χαρακτηριστικών ενός κτιρίου ώστε να είναι εφικτή η κατασκευή του με χρήση τούβλων.
- Τα projects [2] και [3] ασχολούνται με το χτίσιμο τούβλων από ρομποτικά συστήματα. Στην πρώτη περίπτωση χρησιμοποιείται ένας ρομποτικός βραχίονας που χτίζει τούβλα, ενώ στην δεύτερη η τοποθέτηση γίνεται από μικρά ελικοπτεράκια που μεταφέρουν ένα ένα τα τούβλα και τα αφήνουν στις κατάλληλες θέσεις. Αυτές οι εργασίες έχουν ερευνητικό – καλλιτεχνικό χαρακτήρα. Οι παρακάτω εικόνες προέρχονται από αυτές τις δύο εργασίες.

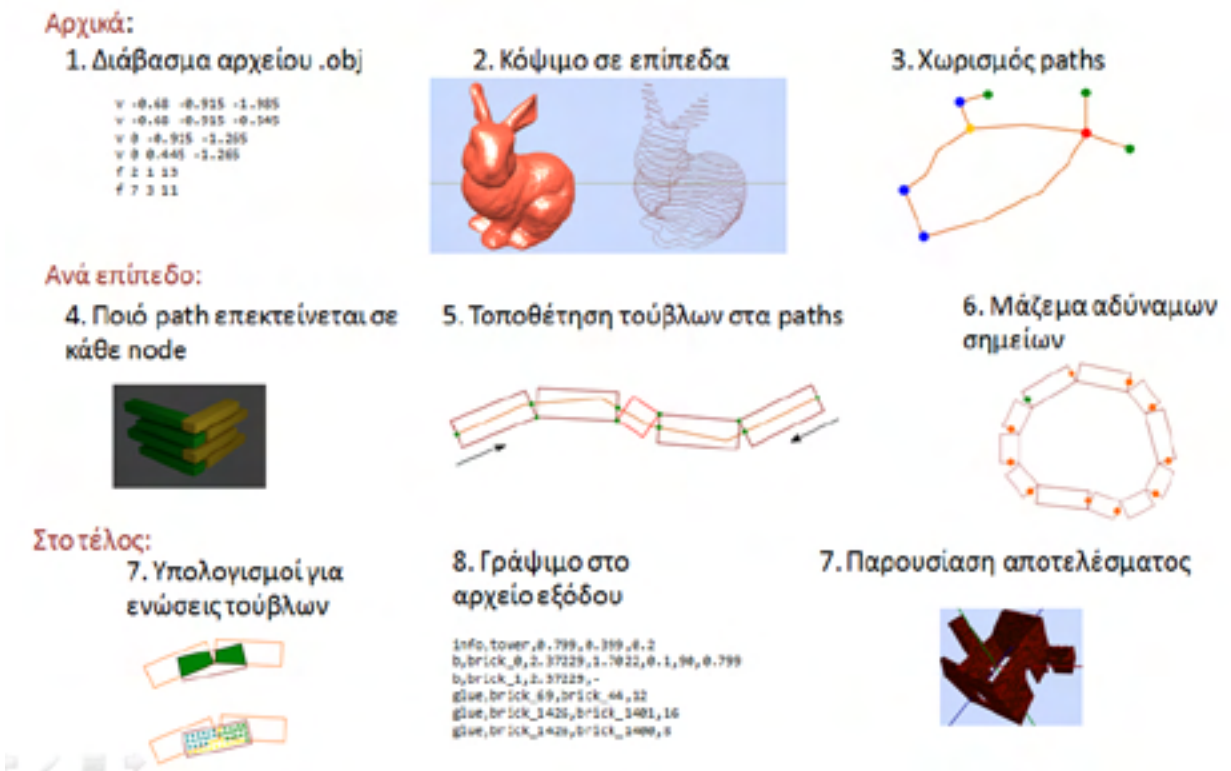
Εικόνα 1: Ρομποτικές διατάξεις χτισίματος τούβλων

- Τα robot Hadrian [5] και SAM [6] αναπτύσσονται με σκοπό τη γρήγορη κατασκευή πραγματικών κτιρίων από τούβλα.

Εικόνα 2: Αριστερά: Fastbrick Robotics Hadrian και δεξιά: Construction Robotics SAM

3. Μεθοδολογία Εικονικού Κτισίματος

Στη παρακάτω εικόνα βλέπουμε συνοπτικά τα βήματα του αλγορίθμου.

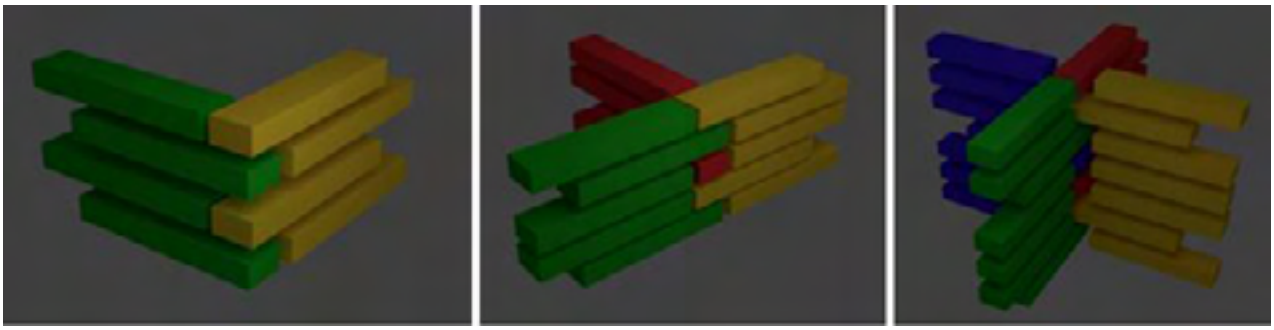


Εικόνα 3: Τα βήματα του αλγορίθμου

Στη αρχή φορτώνουμε το τρισδιάστατο μοντέλο από ένα αρχείο .obj (βήμα 1). Το πρώτο πράγμα που κάνουμε είναι να το «κόψουμε» σε οριζόντια επίπεδα που απέχουν μεταξύ τους όσο το ύψος ενός τούβλου (βήμα 2). Έτσι σε κάθε επίπεδο έχουμε ένα περίγραμμα το οποίο αποτελείται από ευθύγραμμα τμήματα, πάνω στο οποίο θα τοποθετήσουμε τα τούβλα.

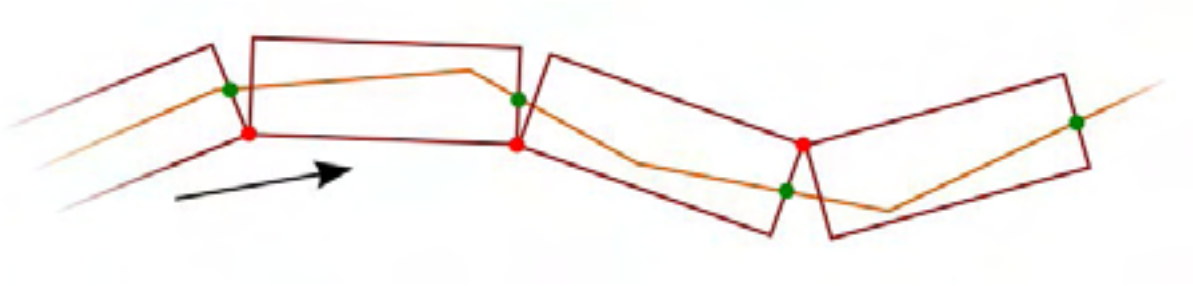
Το επόμενο βήμα είναι ο χωρισμός αυτού του περιγράμματος σε επιμέρους τμήματα ανάλογα με το που εντοπίζουμε γωνίες (βήμα 3). Σα γωνία εδώ θεωρούμε, είτε σημεία στα οποία διασταυρώνονται 3 ή περισσότεροι τοίχοι, είτε σημεία στα οποία η καμπυλότητα του περιγράμματος είναι πολύ μεγάλη. Τα σημεία αυτά τα λέμε nodes και το τμήμα του περιγράμματος μεταξύ δύο nodes το λέμε path. Τα nodes φαίνονται χρωματισμένα στην 3η εικόνα του παραπάνω διαγράμματος.

Στη συνέχεια για κάθε επίπεδο υπολογίζουμε πώς θα γίνει η τοποθέτηση των τούβλων στα nodes ώστε να είναι σωστά πλεγμένα (βήμα 4). Στην παρακάτω εικόνα βλέπουμε πώς γίνεται το πλέξιμο των τούβλων σε nodes με μέχρι 4 τοίχους.



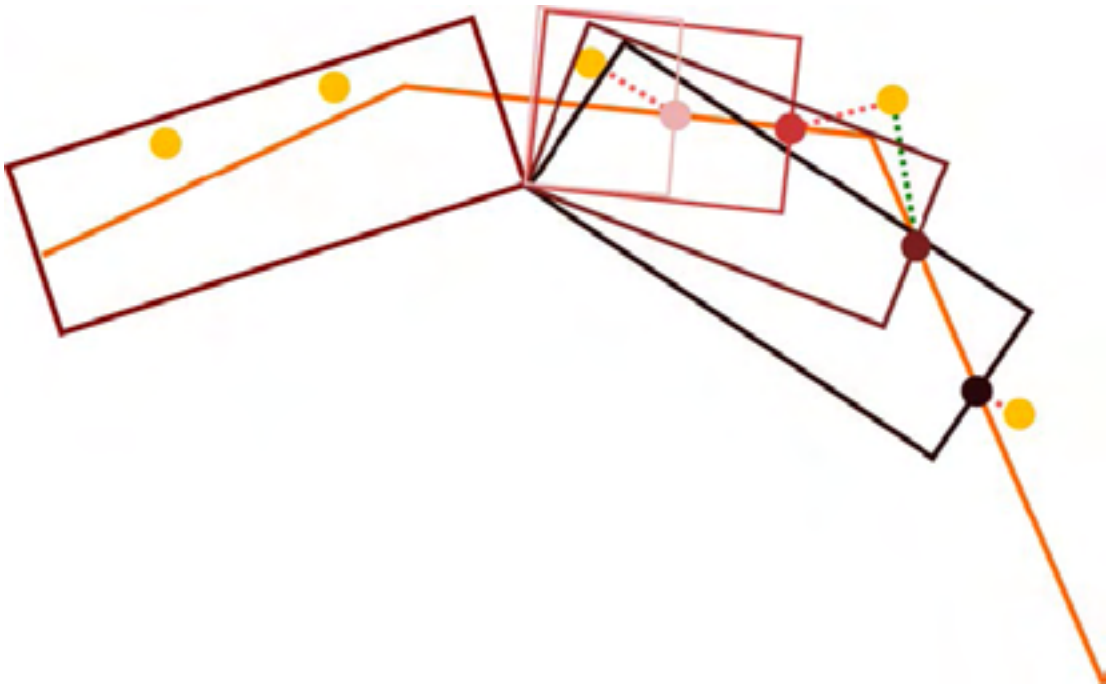
Εικόνα 4: Τοποθέτηση τούβλων σε γωνίες και ενώσεις τοίχων

Αφού ξεκαθαρίσουμε τι γίνεται στα nodes τοποθετούμε και τα εσωτερικά τούβλα στα paths (βήμα 5). Η τοποθέτηση των εσωτερικών τούβλων γίνεται υπολογίζοντας την περιστροφή του τούβλου ώστε να ακολουθεί την πορεία του path. Κάθε καινούριο τούβλο ορίζεται από δύο σημεία. Το ένα ακουμπάει στο προηγούμενό του και υπολογίζουμε τη θέση του δευτέρου, ώστε αυτό να βρίσκεται πάνω στο path. Στην παρακάτω εικόνα βλέπουμε το πρώτο σημείο σημειωμένο με κόκκινο και το δεύτερο με πράσινο.



Εικόνα 5: Τοποθέτηση τούβλων σε ένα path

Επίσης πρέπει να επιλέξουμε ποιο από τα προεπιλεγμένα μήκη θα χρησιμοποιήσουμε για το κάθε τούβλο. Επιλέγουμε αυτό που πρέπει ώστε τα άκρα του τούβλου να βρίσκονται όσο πιο μακρυά γίνεται από τα άκρα των τούβλων του από κάτω επιπέδου, αφού αυτά είπαμε ότι θεωρούμε ως τα αδύναμα σημεία ενός τοίχου και θέλουμε να μην βρίσκονται κοντά μεταξύ τους. Στην παρακάτω εικόνα βλέπουμε τις δοκιμές για τα 4 διαθέσιμα μήκη τούβλου που μπορούμε να χρησιμοποιήσουμε. Για κάθε ένα βρίσκουμε τα σημεία που το ορίζουν. Τελικά εδώ επιλέγουμε αυτό με το 3ο επιτρεπτό μήκος, αφού το τελείωμά του βρίσκεται μακρύτερα από όλα τα αδύναμα σημεία του από κάτω επιπέδου, τα οποία είναι σημειωμένα με κίτρινο.



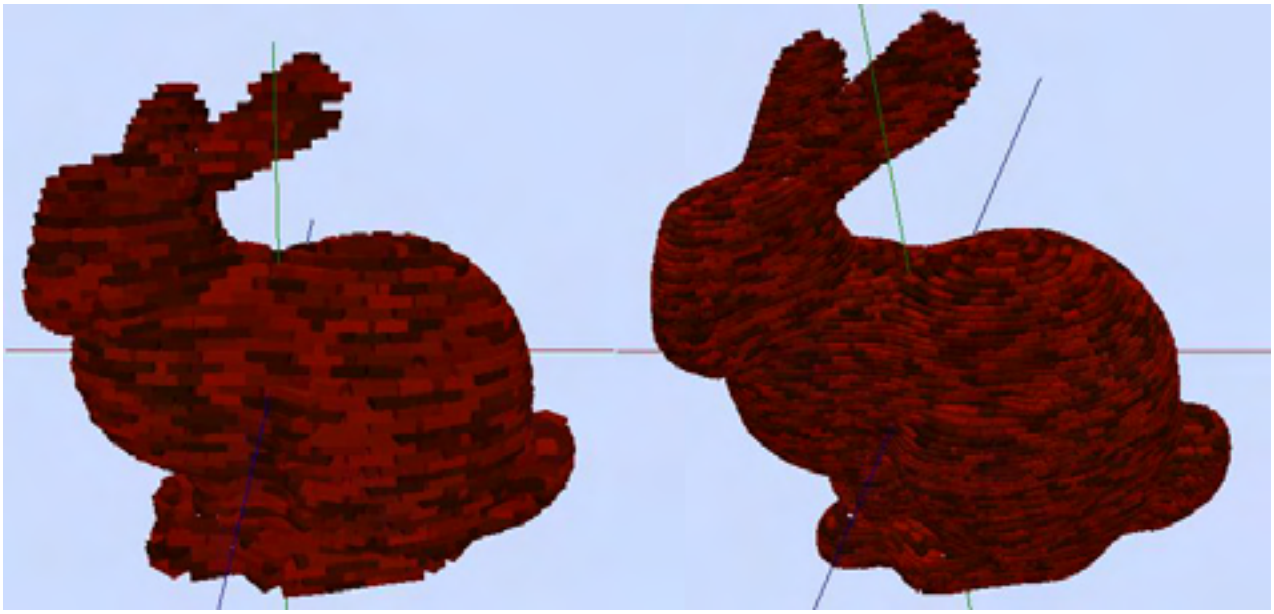
Εικόνα 6: Επιλογή κατάλληλου μήκους για το καινούριο τούβλο

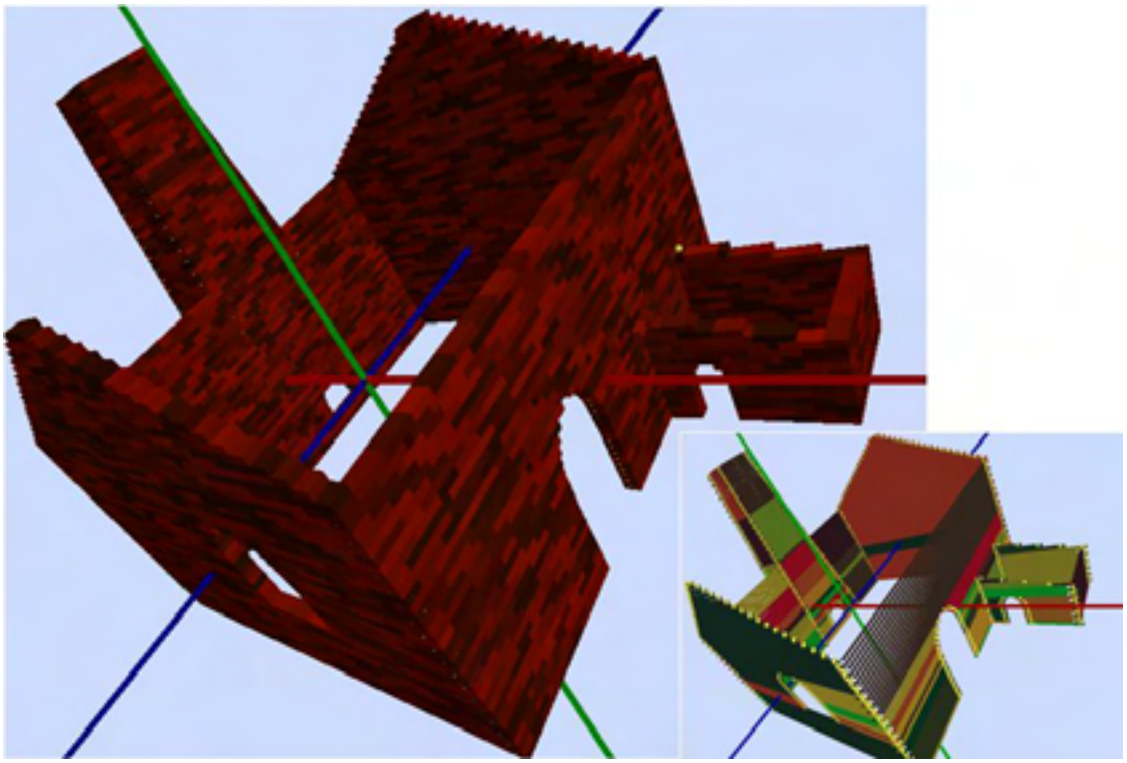
Η τοποθέτηση των εσωτερικών τούβλων δίνει το βέλτιστο αποτέλεσμα ακόμα και με τη χρήση greedy προσέγγισης. Δεν χρειάζεται ποτέ να αναθεωρήσουμε τη θέση κάποιου τούβλου που τοποθετήσαμε. Δεν ισχύει όμως το ίδιο και για τα τούβλα στις γωνίες/ενώσεις τοίχων. Εκεί χρειάζεται κάποια επαναληπτική προσέγγιση για την εύρεση της βέλτιστης τοποθέτησης. Σε αυτό το σημείο βρίσκεται και η μεγαλύτερη αδυναμία της εργασίας μας.

Αφού γίνει η τοποθέτηση των τούβλων μαζεύουμε σε μία λίστα όλα τα αδύναμα σημεία αυτού του επιπέδου γιατί θα μας χρειαστούν κατά την τοποθέτηση των τούβλων της από πάνω στρώσης (βήμα 6).

Αφού ολοκληρωθεί η τοποθέτηση των τούβλων σε όλα τα επίπεδα, βρίσκουμε ποια τούβλα εφάπτονται μεταξύ δύο διαδοχικών επιπέδων και πόσο είναι το εμβαδόν της επιφάνειας επαφής (βήμα 7). Το εμβαδόν αυτό μας δείχνει πόσο ισχυρή είναι η κάθε ένωση. Αυτή η πληροφορία χρησιμοποιείται αργότερα για την εκτέλεση της προσομοίωσης φυσικής της κατασκευής. Χωρίς αυτή την πληροφορία το αποτέλεσμα της προσομοίωσης θα έμοιαζε σαν στοίβα από τούβλα και όχι σαν χτισμένος τοίχος.

Στη συνέχεια αποθηκεύουμε τα αποτελέσματα σε ένα αρχείο .csv (βήμα 8) και εμφανίζουμε το αποτέλεσμα με χρήση OpenGL (βήμα 9). Η παρουσίαση των αποτελεσμάτων φαίνεται στις παρακάτω εικόνες.





Εικόνα 7: Παρουσιάση αποτελεσμάτων

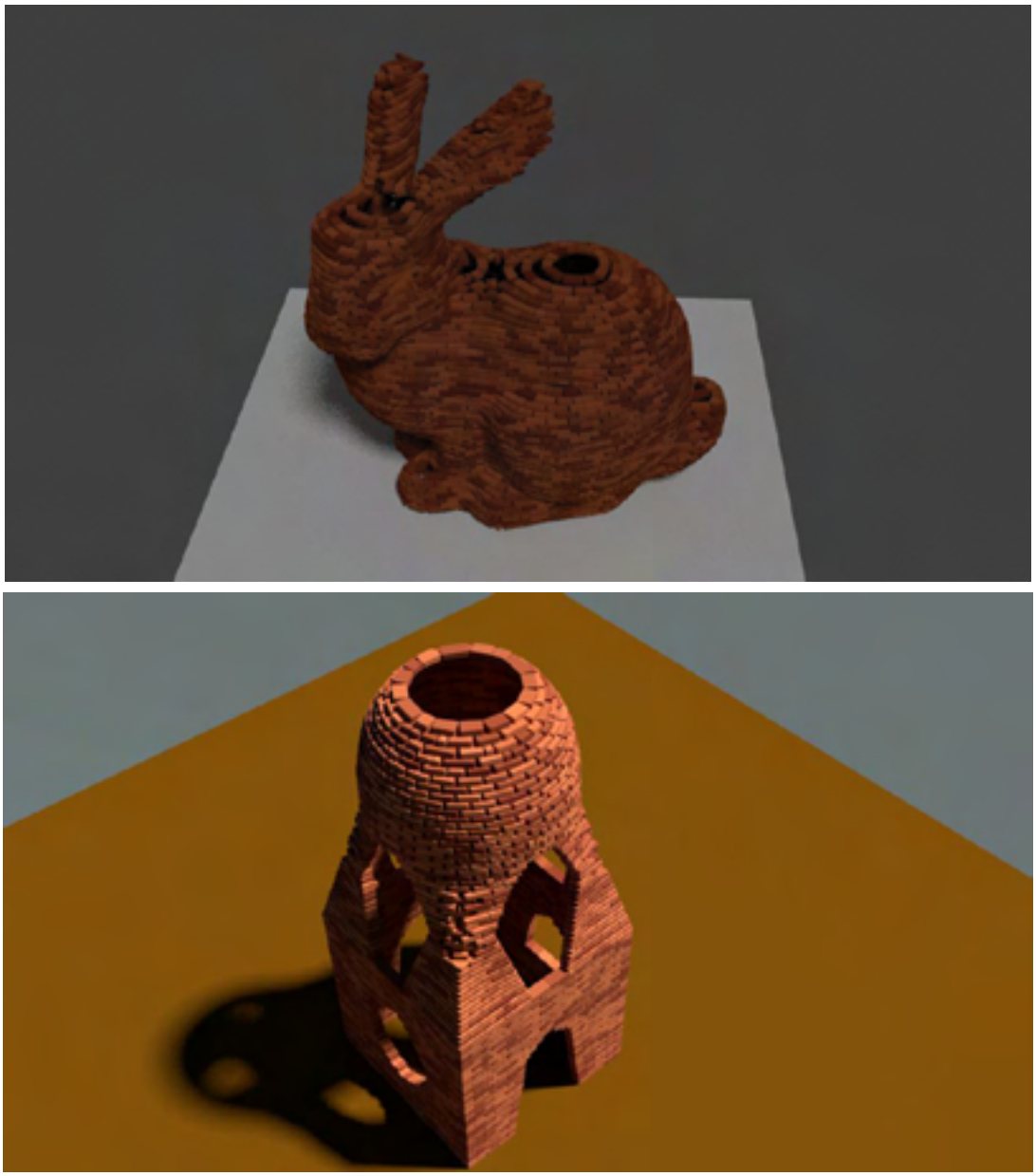
Στην παραπάνω εικόνα, κάτω δεξιά βλέπουμε με χρώματα τα paths που έχουν προκύψει από τους τους τοίχους αυτού του κτιρίου.

4. Προσομοίωση Φυσικής

Για να κάνουμε την προσομοίωση φυσικής της κατασκευής μας, φορτώνουμε τα δεδομένα από το αρχείο .csv στο Blender (ένα Open Source πρόγραμμα δημιουργίας τρισδιάστατων γραφικών). Εκεί υπάρχει ενσωματωμένη η βιβλιοθήκη Bullet Physics η οποία μας δίνει ότι χρειαζόμαστε για την προσομοίωση. Τα τούβλα αναπαρίστανται από Rigid Bodies, με Collision Box σχήματος ορθογωνίου παραλληλεπιπέδου. Οι ενώσεις αναπαρίστανται από Fixed Constraints με ενεργοποιημένη την επιλογή Breakable σε Threshold που εξαρτάται από την ισχύ της κάθε ένωσης. Το φόρτωμα των δεδομένων, η δημιουργία των αντικειμένων και η ρύθμιση των παραμέτρων γίνεται με ένα Python script.

Το Blender εκτός απ' την προσομοίωση φυσικής μας δίνει τη δυνατότητα καλύτερης πιο ρεαλιστικής απεικόνισης των αποτελεσμάτων, όπως φαίνεται στις

παρακάτω εικόνες.



Εικόνα 8: Rendering με χρήση της Cycles Engine του Blender

Παρακάτω φαίνονται κάποια στιγμιότυπα της προσομοίωσης. Στην πρώτη περίπτωση αγνοούμε τις ενώσεις μεταξύ των τούβλων ενώ στη δεύτερη τις χρησιμοποιούμε. Βίντεο της προσομοίωσης υπάρχον στα παρακάτω links:

https://www.youtube.com/watch?v=1Q7dc0E1npU

https://www.youtube.com/watch?v=n22Xt9DX7ZE

Εικόνα 9: Στιγμιότυπα από τις προσομοιώσεις φυσικής

5. Συμπεράσματα

Ψάχνοντας στο Διαδίκτυο μπορεί να βρει κανείς μερικά προγράμματα για την τοποθέτηση τούβλων. Όμως τα περισσότερα μπορούν να χρησιμοποιηθούν σε κατασκευές με πολύ τυποποιημένο σχήμα.

Επίσης υπάρχουν στο Διαδίκτυο αρκετές προσομοιώσεις φυσικής σε κατασκευές από τούβλα. Αλλά στις περισσότερες περιπτώσεις δεν λαμβάνονται υπόψιν οι ενώσεις μεταξύ των τούβλων και οι κατασκευές έχουν κάποιο πολύ απλό σχήμα που συνήθως έχει προκύψει διαδικαστικά.

Το πλεονέκτημα αυτής της εργασίας είναι ότι η τοποθέτηση τούβλων μπορεί να γίνει σε ένα μοντέλο αυθαίρετου σχήματος. Επίσης γίνεται υπολογισμός των ενώσεων μεταξύ των τούβλων, οπότε το αποτέλεσμα της προσομοίωσης ανταποκρίνεται στην πραγματικότητα σε μεγάλο βαθμό.

Σε ένα δημοσίευμα στην εφημερίδα «Ναυτεμπορική» στις 18 Σεπτεμβρίου 2015, αναφέρονται τα εξής:

> “Στα 12 μέτρα ύψος, πρόκειται για τον μεγαλύτερο delta 3D printer στο κόσμο, και είναι ικανός να εκτυπώσει ολόκληρα σπίτια.[...] ο εκτυπωτής είναι κάτι πολύ παραπάνω από την «απλή» πραγματοποίηση ενός ονείρου, αν σκεφτεί κανείς ότι μέχρι το 2030 οι απαιτήσεις όσον αφορά στα σπίτια θα έχουν αυξηθεί κατακόρυφα διεθνώς- με πάνω από 4 δισ. ανθρώπους να ζουν με ετήσιο εισ δημα κάτω των 3.000 δολαρίων. Τα Ηνωμένα Έθνη, αναφέρεται σχετικά, υπολογίζουν ότι μέσα στα επόμενα 15 χρόνια θα υπάρχει η ανάγκη για 100.000 νέα σπίτια ημερησίως. Η ομάδα WASP προτείνει τη συγκεκριμένη επιλογή για σπίτια χαμηλού κόστους, στο πλαίσιο μιας «MakerEconomy», όπου τα πάντα μπορούν να κατασκευαστούν μέσω διαμοιραζόμενων λύσεων. [...]”

Μάλλον το χτίσιμο με τούβλα είναι πιο ταιριαστό για την κατασκευή κτιρίων, απ’ ότι η τρισδιάστατη εκτύπωση. Ουσιαστικά όπως ένας 3D printer αφήνει σταγόνες κάποιου υλικού σε συγκεκριμένες θέσεις, έτσι και ένα ρομπότ που χτίζει, τοποθετεί κάποια τούβλα σε συγκεκριμένες θέσεις. Μπορούμε να πούμε ότι αυτά τα δύο έχουν παρόμοια χρήση, αλλά σε διαφορετική κλίμακα.

Αν ισχύουν τα παραπάνω, τότε ίσως σε κάποια χρόνια να χρησιμοποιούνται

ρομπότ που χτίζουν τούβλα, με σκοπό το μαζικό χτίσιμο κατοικιών. Για να γίνει ο υπολογισμός των θέσεων των τούβλων, θα χρειάζονται προγράμματα παρόμοια με αυτό που αναπτύχθηκε σε αυτή την εργασία.

Αναφορές

[1] E. Whiting, J. Ochsendorf, F. Durand, “Procedural Modeling of Structurally-Sound Masonry Buildings” (SIGGRAPH Asia 09), MIT.

[2] Structural Oscillations, Venice, 2007-2008 Gramazio-Kohler; http://gramaziokohler.arch.ethz.ch/web/e/forschung/142.html [Προσπελάστηκε 9/11/2015]

[3] Flight Assembled Architecture, 2011-2012, FRAC Centre Orléans; http://www.gramaziokohler.com/web/e/projekte/209.html [Προσπελάστηκε 9/11/2015]

[4] E. Lindsay Braley, Brickwork, Pitman, 1963.

[5] Fastbrick Robotics; http://www.fbr.com.au/ [Προσπελάστηκε 9/11/2015]

[6] Construction Robotics; http://www.construction-robotics.com/ [Προσπελάστηκε 9/11/2015]