https://orcid.org/0000-0002-4168-7190
https://orcid.org/0000-0002-3258-7972

UDC 623.764

P.S. SAPATY[*], R. FINKELSTEIN[**]

FIGHTING GLOBAL VIRUSES UNDER SPATIAL GRASP TECHNOLOGY

[*]Institute of Mathematical Machines and Systems Problems National Academy of Sciences of *Ukraine*, Kyiv, Ukraine
[**]Systems Engineering University of Maryland Global Campus, the USA

Анотація. *Робота стимульована жорстокою світовою боротьбою з COVID-19 і є своєрідною спробою взяти участь у цьому глобальному процесі з запатентованою та розробленою мережевою технологією, заснованою на розповсюдженні складних вірусів у великих фізичних та віртуальних просторах. Технологія просторового захоплення (ТПЗ) з базовою Мовою просторового захоплення (МПЗ) використовує паралельний код семантичного рівня, який саморозповсюджується, саморозмножується та самопокривається, створюючи потужні розподілені інфраструктури для вирішення складних проблем. У статті показано, як знайти джерела вірусів у розподілених мережах спочатку шляхом їх відстеження через інфіковані вузли-попередники, якщо такі були зафіксовані, а потім, більш складним чином, шляхом розповсюдження через вузли, що мають менший або близький час зараження, також враховувати можливі збої та динаміку у реальних мережах. Якщо окреслити кількість заражених вузлів, які знаходяться далеко один від одного та з різних сторін зараженої мережі, ймовірне джерело може також знаходитись на перетині найкоротших дерев шляхів, що починаються у всіх таких вузлах, як це також показано на МПЗ. Але аналізуючи складність, динаміку та непередбачуваність реального поширення Covid-19, стає зрозумілою недостатність дискретних мереж для моделювання світового покриття ним. Використовуючи можливість ТПЗ безпосередньо працювати також у фізичних просторах, було показано, як описати глобальний вірус, який розповсюджується масово, причому зараження поширюється за багатьма і навіть досі незрозумілими каналами. У статті також показано, як змоделювати запланований у всьому світі розподіл антивірусної вакцини та її глобальний вплив на поширення вірусу, що символічно представлено як просторова боротьба доброякісного вірусу (вакцина) із шкідливим Covid. Остання версія ТПЗ може бути легко впроваджена та ефективно інтегрована з будь-якими існуючими мережевими системами у глобальному масштабі завдяки встановленню взаємодіючих інтерпретаторів МПЗ від мільйонів до мільярдів копій і таким чином перетворенню усього світу у всемогутню систему для моделювання та управління.*

Ключові слова: *Covid-19, Технологія просторового захоплення, мережеві системи, розподілені обчислення, глобальна пандемія, джерела вірусів, антивірусна вакцина.*

Аннотация. *Работа стимулирована жестокой мировой борьбой с COVID-19 и есть своеобразной попыткой принять участие в этом глобальном процессе с запатентованной и разработанной сетевой технологией, также основанной на распространении мощных вирусов в больших физических и виртуальных просторах. Технология пространственного захвата (ТПЗ) с базовым языком пространственного захвата (ЯПЗ) использует параллельный самораспределяющий, саморозмножающий и самопокрывающий код семантического уровня, создавая при этом мощные распределенные инфраструктуры для решения сложных проблем. В статье показано, как найти источники вирусов в распределенных сетях сначала путем их отслеживания через инфицированные узлы-предшественники, если такие были зафиксированы, а затем, более сложным образом, путем распространения через узлы, имеющие меньшее или более близкое время заражения. Следует также учитывать возможные сбои и динамику в реальных сетях. Если обозначить количество зараженных узлов, которые находятся далеко друг от друга и с разных сторон зараженной сети, вероятный источник может также находиться на пересечении кратчайших деревьев путей, начинающихся во всех этих узлах, как это также показано на ЯПЗ. Но анализируя сложность, динамику и непредсказуемость реального распространения Covid-19, мы поняли недостаточность дискретных сетей для моделирования мирового покрытия. Используя возможность ТПЗ непосредственно работать также в физических пространствах, было показано, как описать глобальный вирус, который распространяется массово, причем заражение распространяется по многим и даже до сих пор непонятными каналам. В статье также показано, как смоделировать запланированное во* *всем мире распределение антивирусной вакцины и ее глобальное влияние на распространение вируса, что символически представлено как пространственная борьба доброкачественного вируса (вакцина) с вредным (Covid). Последняя версия ТПЗ может быть легко внедрена и эффективно интегрирована с любыми существующими сетевыми системами в глобальном масштабе благодаря установлению взаимодействующих интерпретаторов ЯПЗ от миллионов до миллиардов копий и таким образом превращению всего мира во всемогущую систему для моделирования и управления.*

Ключевые слова: *Covid-19, Технология пространственного захвата, сетевые системы, распределенные вычисления, глобальная пандемия, источники вирусов, антивирусная вакцина.*

Abstract. *This has been inspired by the current world fight with COVID-19, in an attempt to participate in it with patented and developed networking technology, also based on spreading powerful viruses in large physical and virtual spaces. The Spatial Grasp Technology (SGT) with basic Spatial Grasp Language (SGL) is using parallel self-spreading, self-replicating, and self-matching semantic level code creating powerful distributed infrastructures for solving complex problems. The paper shows how to find virus sources in distributed networks, first, by tracing them via infected predecessors if such were fixed, and then, more complexly, by moving through nodes with lower or close infection time, also taking into account possible failures in real networks. If to outline a number of infected nodes staying far away from each other and on different sides of the infected network, the probable source may also be on intersection of shortest path trees starting in them, as shown in SGL. But analyzing complexity, dynamics, and unpredictability of spread of Covid-19, we understood the insufficiency of discrete networks for simulating its world coverage. By using the SGT capability to directly operate in continuous physical spaces too, we showed how to describe the global malicious virus in a massive way, with the infection spreading via many and so far unclear channels. The paper also shows how to model the planned distribution of the antivirus vaccine and its global impact on the virus, symbolically presented as spatial fight of benign (vaccine) with malicious (Covid) viruses. The latest version of SGT can be implemented and integrated with any existing networked systems in a global manner, with installment of communicating SGL interpreters in millions to billions copies and converting the world into a global simulation and control engine.*

Keywords: *Covid-19, Spatial Grasp Technology, networked systems, distributed computing, global pandemics, virus sources, antivirus vaccine.*

DOI: 10.34121/1028-9763-2020-4-113-124

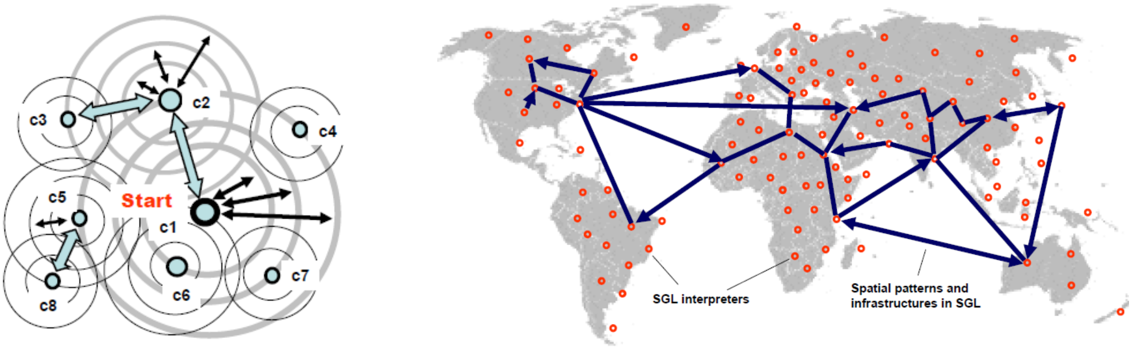
1. Introduction

This work has been inspired by the current fight with COVID-19 [1–13], as an attempt to participate in this global process with the patented, developed, and tested in different countries high-level networking ideology and technology also based on spreading powerful viruses in large physical and virtual spaces [14–22]. Such technology, except parallel wavelike coverage of distributed spaces, dynamically creates powerful distributed infrastructures capable of defeating other models and believably any global viruses too. Detailed analysis of how Covid-19 spreads and influences the world shows that effective fighting with it can only be done by intelligent spatial technologies capable of understanding and grasping the world as a whole, having both global control and detailed live access to local data scattered throughout numerous world points, which may constantly change in time. And these absolutely necessary conditions stimulated us to investigate SGT for the global virus fight, as possessing the desired features which have been already tested on numerous applications in different civil and defense areas. The latter even linked with sociology and psychology (like formalizing gestalt psychology laws for their capability of operating in large distributed systems and not only in localized human brain), and most recently, in simulating global awareness and a sort of distributed consciousness as top level of natural and artificial intelligence, with potential applications in massive and swarm robotics.

The rest of this paper is organized as follows. Section 2 briefs the developed SGT based on self-spreading, self-replicating, and self-modifying higher-level code covering and matching distributed systems at runtime, while providing global integrity, goal-orientation, and effective solutions. Section 3 shows how to model spreading virus in distributed networks and trace its source for any infected node via the infected predecessors if such are registered. Section 4 describes a more complex situation where the virus source is trying to be found by knowing only infection time in nodes with also taking into account real network dynamics and existence of failed components. Section 5 shows an attempt of defining probable virus source lying on intersection of shortest path trees from a set of selected infected nodes, which can generally result in a number of solutions. Section 6, after examining the records on Coivid-19 worldwide spread, comes to conclusion of insufficiency of traditional network models to simulate and explain such pandemics, and using SGT capability to directly operate in distributed physical spaces too, shows a sketch in SGL of how to model its massive spread via numerous and so far not fully understood channels. Section 7 reviews world attempts and existing problems to create and distribute antivirus vaccine and describes how to model its world distribution in SGL, also showing its spatial fight with the simultaneously spreading Covid. Section 8 concludes the paper mentioning the possibility of quick implementation of the latest technology version even in traditional university environments, also hinting on further research plans for the use of SGT for creating international technological basics capable of fighting forthcoming pandemics and other global disasters.

2. Spatial Grasp Technology Basics

We are providing here only main ideas of SGT on which there exist available sources including European patent, Wiley, Springer and Emerald books, as well as other publications and presentations of different technology versions with networked implementations and diverse applications, with only some at [14–22]. Within SGT, a high-level scenario for any task to be performed in a distributed world is represented as an active self-evolving pattern rather than traditional program, sequential or parallel. This pattern, written in a high-level Spatial Grasp Language (SGL) and expressing top semantics of the problem to be solved, can start from any world points. It then spatially propagates, replicates, modifies, covers and matches the distributed world in parallel wavelike or even virus-like mode, while echoing the reached control states and obtained data (which may be arbitrarily remote) for making decisions and processing at higher levels, with further space navigation from the new or previously reached points, and so on. This parallel and distributed world spreading-covering-conquering-matching-grasping-echoing process is symbolically shown in Fig. 1.



a *b*

Figure 1 – Basic ideas of SGT: (a) controlled wavelike space navigation & matching; (b) parallel world coverage

Many spatial processes in SGL can start any time and from any world points and develop simultaneously, by cooperating or competing with each other. The self-spreading and matching SGL patterns can create knowledge infrastructures arbitrarily distributed between system components having embedded SGL interpreters, which may cover any regions, the whole world including, as in Fig. 1 *b*. The created infrastructures, which may be left active, can effectively support or express distributed databases, command and control, situation awareness, autonomous decisions, and any other existing or hypothetical computational and control models (like, say, neural networks or Petri nets).

SGL has a recursive structure with its top level shown below, which reflects the space-grasping-creation-matching-echoing nature of SGL scenarios (with words in italics representing syntactic categories, square brackets showing optional constructs, braces indicating repetitive parts, and parentheses and comma as the language symbols).

grasp → *constant* | *variable* | [*rule*] [({ ***grasp*** **,** })]
constant → *information* | *matter* | *custom* | *special* | ***grasp***
variable → *global* | *heritable* | *frontal* | *nodal* | *environmental*
rule → *type* | *usage* | *movement* | *creation* | *echoing* | *verification* | *assignment* |
advancement | *branching* | *transference* | *exchange* | *timing* | *qualifying* | ***grasp***

The SGL scenario can dynamically spread & process & match the world or its parts needed, with scenario code capable of virtually or physically splitting, replicating, modifying and moving in the distributed spaces being accompanied with transitional data in special frontal variables. This movement can take place in single or multiple scenario parts spatially interlinked under the overall control, which is spreading, covering, evolving and matching the navigated world too.

Numerous communicating SGL interpreter copies can be installed worldwide in thousands to millions to billions of communicating copies and integrated with any other systems, while forming altogether a sort of global spatial brain governed on top level by freely moving consciousness-like scenarios in SGL.

3. Tracing Virus Source via Infected Predecessors in Networks

Finding virus sources in networks is generally a complex problem with many existing publication, with only some by [23–26]. We describe here only a very simplified case where the spread of biological or computer virus in a distributed social or computer network is supposedly fixed in all infected nodes, and its source can be traced by stepwise movement via the nodes from which the current nodes got the virus, as shown in Fig. 2.

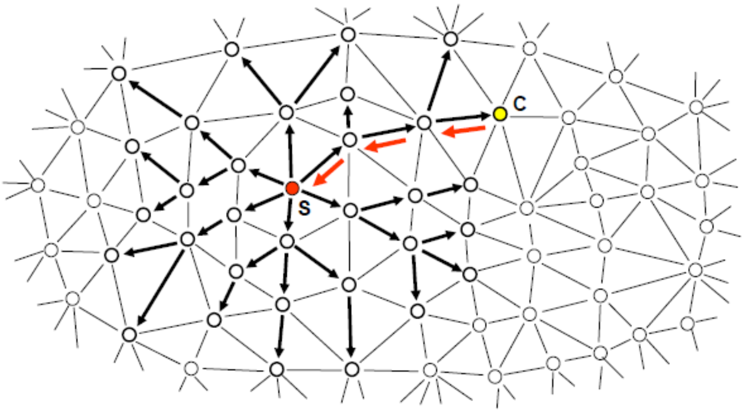


Figure 2 – Spreading virus and tracing its source from an infected node

A stepwise parallel propagation of a virus in a network from its source `S` to a certain depth (symbolically 100 steps) can be organized in SGL as follows, with nodes referring to predecessor nodes that infected them in nodal variables `Before`. Repeated parallel invocation of the mobile code starts in each new node reached, and nodes are allowed to be entered only once, to prevent looping.

```
nodal(Before); hop_first(S);
repeat_100(hop_first(links(all)); Before = BACK)
```

Starting from any infected node (let it be `C`) and tracing virus source via the infected predecessors by mobile SGL code may be as follows, with this spatial cycle terminating in the virus source node having no registered infection predecessor and its name issued outside of the network.

```
hop_direct(C); repeat(hop(link(any), node(Before))); output(NAME)
```

The output will be: S

4. Finding Probable Virus Source via Infection Time in Nodes

Finding virus sources is generally not a trivial task as, for example, discussed in [23, 24], being much more complicated than the simplified example of the previous section, as different nodes may generally not know which particular neighbors infected them. We assume here that spreading viruses are registered somehow in network nodes only by the time there were infected, and will try to use such time records to trace the virus source. Parallel spreading of a virus from its source `S` with registering infection time in the reached nodes can be expressed as follows, where environmental variable `TIME` in SGL is capable of accessing the absolute world time.

```
hop_first(S); nodal(Infected) = TIME;
repeat_100(sleep(delay); hop_first(links(all)); Infected = TIME)
```

For a pure distributed simulation example, we can use the growing virtual system time in the frontal variable `Time`, registered in nodes by nodal variable `Infected`, with its initial value *`start`* in the virus source node, which is regularly increasing by *`delay`* value in nodes before reaching new nodes in parallel, and so on, as follows.

```
hop_first(S); frontal(Time) = start; nodal(Infected) = 0;
repeat_100(Time += delay; hop_first(links(all)); Infected = Time)
```

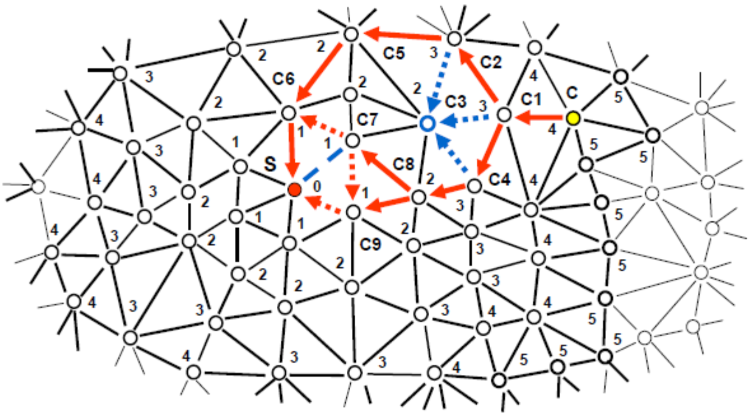


Figure 3 – Finding probable virus source by known node infection times

We are considering finding possible virus source by starting from some infected node (`C`, as in Fig. 3, with infection time shown at each node) and assigning the registered infection time in nodal variable `Infected` to the moving (frontal) variable `Duration`, which will be updated each time when arriving in new nodes. We can also assume that in large real networks some links between nodes, the whole nodes too, may happen to be non-operational or even broken, so we decide to move in the network not only to nodes with lower infection time, but if not possible, also with the same, approximate or even higher infection times, in hope to ultimately reach nodes with the lower time values. The following scenario reflects these ideas, and by carrying the reducing value in `Duration`, tries to finally reach the node or nodes with the lowest infection time, which may hopefully be the virus source, or at least close to it, if the latter not physically reachable.

```
frontal(Duration); nodal(infected);
hop_first(C); Duration = Infected;
output_min(
  repeat(
    or_seq(hop_first(links(any), nodes(Infected < Duration)),
           hop_first(links(any), nodes(Infected == Duration)),
           hop_first(links(any), nodes(Infected > Duration))));
  Time && NAME)
```

Some explanation for the situation in Fig. 3 is as follows. Starting from node `C` with infection time 4, the scenario first finds the only neighboring node `C1` with lesser time, 3, from which it was not possible to move further with smaller time, as the only node `C3` (in blue) with time 2 appeared damaged. The scenario then moves in parallel to two neighbors `C2` and `C4` with same time 3, from which sequences (`C2`, `C5`, `C6`, `S`) and (`C4`, `C8`, `C9`, `S`) with reduced time at each step are possible, with additional chains (`C4`, `C8`, `C7`, `C6`, `S`) and (`C4`, `C8`, `C7`, `C9`, `S`) possible too if to move to neighbors with same infection time without other options (like due to broken line, in blue, between `C7` and `S` nodes). This scenario can finally come to a single node with smallest infection time which may be the virus source, like `S`, or to others, and more than one, with the smallest detected infection time `S` appeared not available. The output issued outside of the network will be a couple containing the value of smallest infection time found and the name of related infected node, which may represent the virus source or other nodes maximally close to the source.

5. Finding Virus Source on Intersection of Shortest Path Trees from Selected Nodes

If to outline a number of infected nodes staying far away from each other and presumably on opposite sides of the infected network (for which some preliminary hints should exist or analyses made on the network's extent and general structure), the probable source may be expected in the point from which shortest paths exist to all selected nodes, assuming the virus spread with same speed throughout the network. This may technically mean that the infection source could be found on intersection of shortest path trees starting in all selected nodes, as shown Fig. 4.

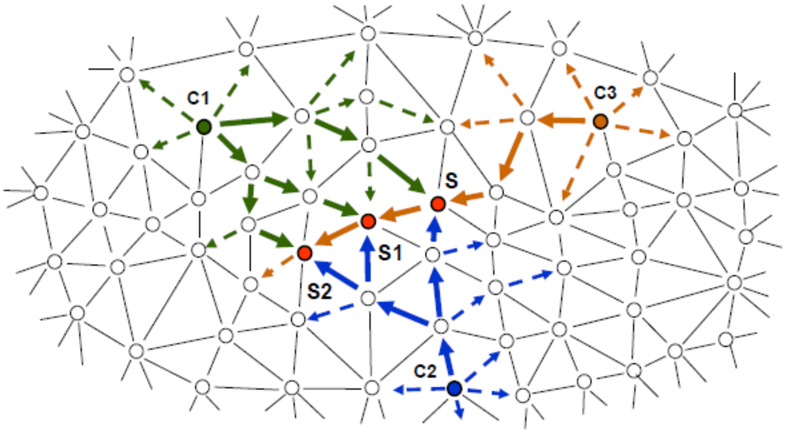


Figure 4 – Fining possible source as having shortest paths to all selected nodes

The following SGL scenario, starting in parallel from the selected nodes `C1`, `C3`, and `C3`, finds shortest path trees (SPT) from them restricting their depth (or height) in `Limit`, as the network can potentially be very large (other useful restriction may be by moving only via infected nodes, if such are registered, in hope that the infection covers only part of the network). Different SPT are registered in nodes by variables `Back` naming right above standing node in the tree, with such variables in same nodes for different trees having colors (using moving variables `IDENTITY`) of the corresponding trees reflecting names of their starting (or root) nodes. After getting all SPT, the scenario again starts in parallel from their tope nodes and moves down only to nodes having registered predecessors, adding to them the color of this tree. And when a node collects colors of all trees, it is issued as a probable infection source. But in general, the intersection of all SPT may hint to more than a single potential virus source, as shown in Fig. 4 (with `S`, `S1`, and `S2`), say, when the physical speed of virus varied in different directions in the network, so the different obtained sources should be further examined on their probability.

For this, during the ascending through SPT, the scenario also carries `Distance` value from the tree root to each reached node and summarizes it for all SPT covering this node, issuing it together with the probable source name. For example, the node with minimum sum can be more probable to be the virus source if it spread in same or similar speed throughout the network, and especially if the infection time in all selected nodes (if registered) was same or close to each other's (in Fig. 4, for this case node `S` would be the most probable source). If infection times in such nodes differed essentially, other nodes (like `S1` or `S2`, or more) could be the source too. Further investigation of possible relation of infection times in selected nodes with obtained sums of shortest paths to them from a probable source could be worth of conducting too.

```
sequence(
 (hop(C1, C2, C3);
  IDENTITY = NAME; nodal(Back, Far);
  frontal(Length, Limit = number);
  repeat(
    hop_links(all); Length += 1;
    if(or(Far = undefined, Far > Length),
        (Far == Length; Back == BEFORE), stop);
    Length < Limit)),
 (hop(C1, C2, C3);
  frontal(Distance) = 0; nodal(Common, Sum); IDENTITY = NAME;
  repeat(
    Sum += Distance; append(IDENTITY, Common);
    if(count(Common) == 3, (output(NAME, Sum); Common = 0));
    Distance += 1; hop(links(all), nodes(Back = BEFORE)))))
```

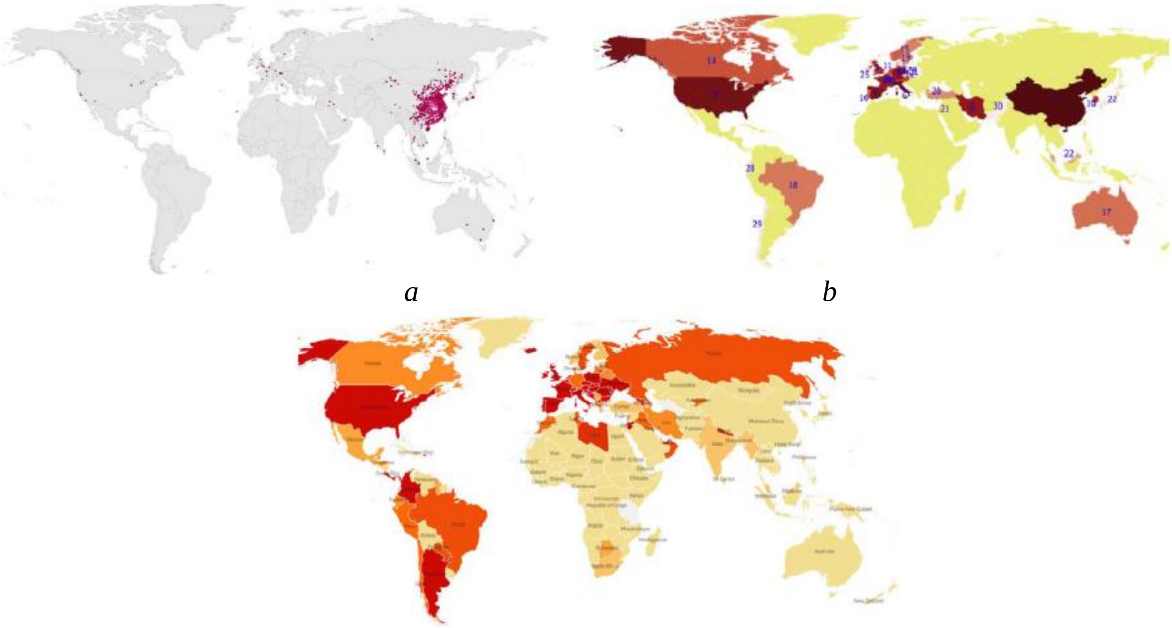
The output of possible virus sources together with sums of shortest paths from them to the selected nodes will be as: (S, 9), (S1, 10), (S2, 11), …

In a very simplified way, if to assume that SGL interpretation operates on a real network with topology similar to Fig. 4, and its code spreads in parallel through the network links similar to what real virus did, we may have the spatial simulation of the virus spread with its source location as follows (with source `S` in Fig. 4 as the most probable solution).

```
hop_first(C1, C2, C3); nodal(Count);
repeat(hop_first_links(all); Count += 1;
        If(Count == 3, (output(NAME); abort)))
```

6. Massive Spread of Coronavirus and Its Possible Simulation with SGT

But analyzing the complexity and dynamics of the spread of Covid-19 [1–6], we understood the insufficiency of discrete networks for simulating its world coverage, as the virus looks like spreading via multiple, most diverse, and not completely understood channels, from personal communications to possibly just by the air too. In Fig. 5 *a-c*, consecutive stages of its world coverage are shown for different periods of time (as from [1–3]), with darker areas indicating higher infection intensities, which do not hint on clear ideas of the direction, breadth and depth of its spread, which looks chaotic and unpredictable.



a *b*

c

Figure 5 – Covid-19 world coverage: a) December 2019 to February 2020; b) by March 2020; c) by October 2020

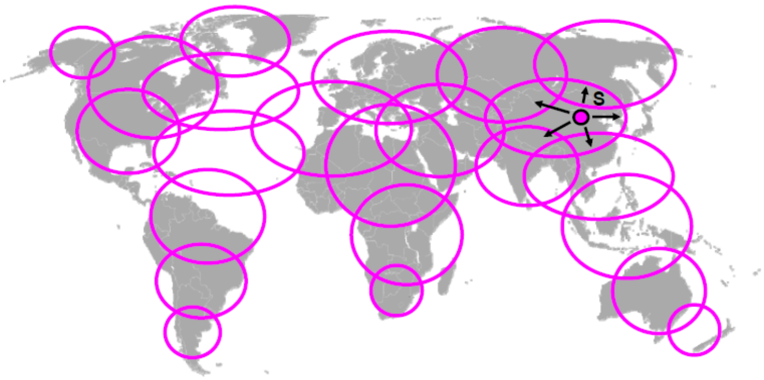


Figure 6 – Simulating global spread of virus in physical space

By using the SGT capability of directly staying and operating in continuous physical spaces too, not only in discrete networked structures, we can describe the global virus spread in the most massive and diverse way, with the infection spreading via many and so far unknown channels. For example, to express in SGT, which itself is a super-virus ideology and technology, the virus originating and spreading randomly worldwide in both breadth and depth mode, with capability of return to the previous regions, we may just write the following (with Fig. 6 symbolically reflecting its spatial activity).

```
move(S_coordinates); nodal(Status);
frontal(Breadth = number, Limits = (Xmin_Xmax, Ymin_Ymax));
Status = infected;
repeat(
  parallel_Breadth_copy(shift(random(Limits)));
  Status == nil; Status = infected)
```

This scenario generally provides parallel stepwise and randomized movement in physical space to any extent, where reaching each point may be linked with influencing resources in these locations corresponding to individuals or their collectives up to whole countries, depending on the needed details of simulation. The chosen details may also depend on the existing physical capability of world coverage by the spreading recursive SGL code and the number of installed SGL interpreters, which may well be in millions to billions and deeply integrated with any existing systems, the whole internet including. The above scenario may also prove to be a starting skeleton for further virus modeling and simulation, as well as real management for dealing with pandemics-like disasters, which are planned to be addressed in the following research and publications.

7. Distribution and Influence of Antivirus Vaccine

Great efforts in different countries of the world are taking place to develop suitable Covid-19 vaccine, test it, produce massively, and make world-wide distribution [7–13], with enormous amount of problems arising, however. World Health Organization (WHO) has pushed countries to sign up for a plan that will buy a vaccine in huge quantities and distribute it in an equitable way [7]. But it has been grappling with two big issues: how to get high-income countries to join, instead of hoarding early vaccine supplies for their own populations; and how to share the vaccine in a fair way once it becomes available. Also, how to distribute the developed vaccine around the globe, where on preliminary estimates, the equivalent of 8,000 Boeing 747s will be needed, as the International Air Transport Association (IATA) has said [13]. We also need being prepared for a next pandemic, which may well happen too. In this short paper we cannot address such problems in any detail, but are trying to show at least some principle of how they can be modeled and managed by intelligent spatial control like the one provided by SGT.

Unlike the malicious virus spreading worldwide massively, chaotically, and unpredictably, the distribution of vaccine can be organized in a much more directed and controllable way. Of course, this should also take into account emerging peculiarities of different regions and their remoteness, also allowing for certain degree of randomness, as destinations and local distribution conditions may not be fully clarified in advance.

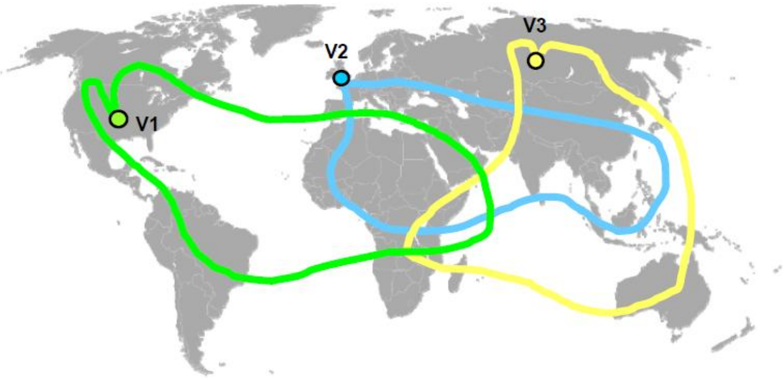


Figure 7 – World distribution of antivirus vaccine

The following SGL scenario, starting in nodes V1, V2, and V3 (see Fig. 7) with accumulated initial amount of vaccine doses (here for simplicity, same in each) distributes these doses in physical space in a combined breadth-depth mode. From each reached point, with currently available vaccine in frontal variable Amount, the next Breadth number of branches is defined and pursued randomly chosen destinations in each in parallel. The available amount of vaccine propagating with each branch will correspond to Amount divided by Breadth which will be moving further in parallel too, within spatially replicated variables Amount. In each reached point, a Pack number of doses needed for local consumption will be withdrawn, and so on, unless the amount of propagating vaccine is finally exhausted. The points (or regions) to which the Pack of vaccine is delivered are acquiring status protected and become unreachable for the spreading virus.

```
move(V1, V2, V3); nodal(Status);
frontal(
  Start = WHERE, Expanse = 0, Breadth = number, Shift, Amount = all_doses,
  Limits = (Xmin_Xmax, Ymin_Ymax), Distance, Pack = consumed_doses,
  Min = min_dose);
Status = protected;
repeat(
  Access = register(
```

```
     parallel_Front_copy(
       Shift = random(Limits);
       Distance = distance(Start, WHERE + Shift);
       Distance > Expanse; Expanse = Distance;
       shift(Shift); no(Status == protected));
  Amount = Amount / count(follow(Access));
  follow(Access); decrement(Amount, Pack) > Min;
  Status = protected)
```

This vaccination scenario can operate in distributed spaces simultaneously with the previous infection scenario, but assigning to the accessed and covered regions the status `protected` it blocks further development of coronavirus in (and from) these points and areas. For such parallel and competitive operation the previous infection scenario should have some minor updates, as follows.

```
move(S_coordinates); nodal(Status);
frontal(Breadth = number, Limits = (Xmin_Xmax, Ymin_Ymax));
Status = infected;
repeat(
  parallel_Breadth_copy(shift(random(Limits)));
  notbelong(Status, (infected, protected)); Status = infected)
```

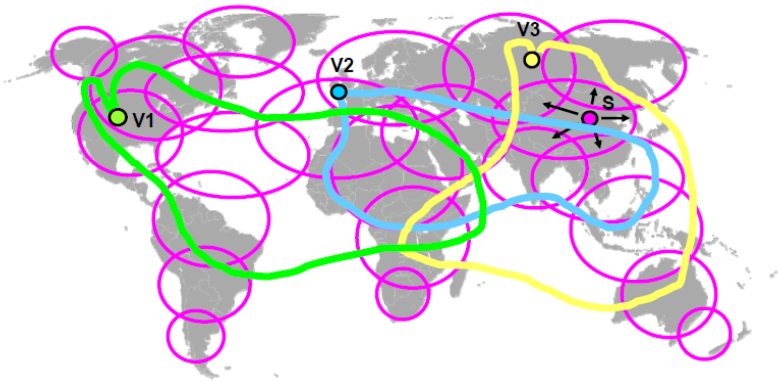


Figure 8 – Spatial fight of distributing vaccine with the spreading virus

Such parallel distributed simulation of actually two spreading and fighting each other viruses, with one being malicious (Covid) and the other benign (vaccine), may be extremely interesting and useful for effective withstanding and ultimate removal of the current coronavirus, also for creation of advanced methodologies and technologies for analyzing, fighting and predicting of any future pandemics.

8. Conclusions

The paper investigated the possibility of using developed Spatial Grasp model and technology for solving worldwide problems linked with the spread and influence of global epidemics. Originally based on the self-spreading semantic level virus it has a great power for both creating and supporting spatial systems, as was extensively investigated and published, but also possess enormous capabilities for defeating and destroying other systems and organizations, among which may be the current Covid. And this paper is declaring its challenge to fight such global disasters, which may relate not only to pandemics but also to climate change, different world conflicts, also existing and growing problems in global security and defense. New publications are planned in this area, including the new book which is currently in completion. The latest SGL version can be quickly implemented even within standard university environments, similar to its previous versions in different countries under the author's supervision. The technology can be installed in numerous copies worldwide and deeply integrated with any other systems, overall internet including, actually acquiring unlimited power for simulation and management of the whole world.

REFERENCES

1. Xu B., Gutierrez B., Mekaru S. Epidemiological data from the COVID-19 outbreak, real-time case information. *Sci Data*. 2020. Vol. 7, N 106. URL: https://www.nature.com/articles/s41597-020-0448-0.
2. Mandal C.C., Panwar M.S. Can the summer temperatures reduce COVID-19 cases? *Public Health*. Aug. 2020. Vol. 185. P. 72–79. URL: https://www.sciencedirect.com/science/article/pii/S0033350620302304?via%3Dihub.
3. Covid World Map: Tracking the Global Outbreak. *By The New York Times* (Updated, October 25). 2020. 12:33 A.M. E.T. URL: https://www.nytimes.com/interactive/2020/world/coronavirus-maps.html.
4. Woodie A. Tracking the Spread of Coronavirus with Graph Databases. *Datanami*. 2020. March 12. URL: https://www.datanami.com/2020/03/12/tracking-the-spread-of-coronavirus-with-graph-databases/.
5. Tracking the spread of the novel coronavirus. 2020. Oct. 29. URL: https://graphics.reuters.com/CHINA-HEALTH-MAP/0100B59S39E/index.html.
6. Kupferschmidt K. WHO unveils global plan to fairly distribute COVID-19 vaccine, but challenges await. *Science*. 2020. Sep. 21. URL: https://www.sciencemag.org/news/2020/09/who-unveils-global-plan-fairly-distribute-covid-19-vaccine-challenges-await.
7. Gorelick K. Here’s how hard it will be to distribute a coronavirus vaccine. *The Washington Post*. 2020. September 25. URL: https://www.washingtonpost.com/outlook/2020/09/25/covid-vaccine-distribution-logistics/.
8. The push for a COVID-19 vaccine. World Health Organization. URL: https://www.who.int/emergencies/diseases/novel-coronavirus-2019/covid-19-vacines?gclid=Cj0KCQjwxNT8BRD9ARIsAJ8S5xZW86w1vwoZ4sFhM1wPctl_322RetSbTVrpLeM88Pzt3oztC-jX3kYaAuPsEALw_wcB.
9. Palca J. Can You Tell If A COVID-19 Vaccine Is Working? *HEALTH NEWS FROM NPR*. 2020. September 7. URL: https://www.npr.org/sections/health-shots/2020/09/07/909713885/how-can-you-tell-if-a-covid-19-vaccine-is-working.
10. Sagonowsky E. Planes, trucks and ultracold boxes: Pfizer preps massive COVID-19 vaccine distribution effort. 2020. Oct 21. URL: https://www.fiercepharma.com/manufacturing/pfizer-designed-new-container-and-plans-to-tap-shipping-companies-for-covid-19.
11. Theobald N. Emerging vaccine delivery systems for COVID-19. *Drug Discov Today*. 2020. Vol. 25, N 9. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7313501/.
12. Covid vaccine: 8,000 jumbo jets needed to deliver doses globally, says IATA. *BBC News*. 2020. September 10. URL: https://www.bbc.com/news/business-54067499.
13. Sapaty P. A Distributed Processing System: European Patent. № 0389655; Publ. 10.11.93; European Patent Office, Munich, 1993.
14. Sapaty P.S. Mobile Processing in Distributed and Open Environments. New York: John Wiley & Sons, 1999. 410 p.
15. Sapaty P.S. Ruling Distributed Dynamic Worlds. New York: John Wiley & Sons, 2005. 255 p.
16. Sapaty P.S. Managing Distributed Dynamic Systems with Spatial Grasp Technology. Springer, 2017. 284 p.
17. Sapaty P.S. Holistic Analysis and Management of Distributed Social Systems. Springer, 2018. 234 p.
18. Sapaty P.S. Complexity in International Security: A Holistic Spatial Approach, Emerald Publishing, 2019. 160 p.
19. Sapaty P.S. Mosaic Warfare: From Philosohpy to Model to Solution. Mathematical Machines and Systems. 2019. Vol. 5, N 5. URL: https://medcraveonline.com/IRATJ/IRATJ-05-00190.pdf.
20. Sapaty P.S. Advanced terrestrial and celestial missions under spatial grasp technology. *Aeronautics and Aerospace Open Access Journal*. 2020. Vol. 4, Iss. 3. URL: https://medcraveonline.com/AAOAJ/AAOAJ-04-00110.pdf.
21. Sapaty P. Symbiosis of Real and Simulated Worlds Under Global Awareness and Consciousness. Abstract at*: The Science of Consciousness Conference*. 2020. Sept. 14–18. URL: https://eagle.sbs.arizona.edu/sc/report_poster_detail.php?abs=3696.
22. Shah D., Zaman T. Detecting Sources of Computer Viruses in Networks: Theory and Experiment. *SIGMETRICS’10*. June 14–18. 2010. New York, USA. URL: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.208.3857&rep=rep1&type=pdf.

23. Huang Q. Source Locating of Spreading Dynamics in Temporal Networks. *Proc. of the 26th International Conference on World Wide Web Companion*. 2017. April. P. 723–727. URL: https://dl.acm.org/doi/abs/10.1145/3041021.3053376?download=true.

24. Pinto P.C., Thiran P., Vetterli M. Locating the source of diffusion in large-scale networks. *Phys.rev.lett*. 2012. Vol. 109, N 6. P. 1–5. URL: https://www.researchgate.net/publication/230657651_Locating_the_Source_of_Diffusion_in_Large-Scale_Networks.

25. Shen Z., Cao S., Fan Y., Di Z., Wang W.X., Stanley H.E. Locating the source of spreading in complex networks. *Computer Science*. 2015. URL: https://www.researchgate.net/publication/271448247_ Locating_the_source_of_spreading_in_complex_networks.

Стаття надійшла до редакції 13.11.2020