Comparison of a Physically Inspired Agent-Based Model with a Simulated Annealing Algorithm for the Vehicle Routing Problem

Kerstin Lenger
FH JOANNEUM – University of Applied Science,
Data Science and Artificial Intelligence
8020 Graz, Austria
kerstinlenger@gmx.at

Klaus Lichtenegger
FH JOANNEUM – University of Applied Science,
Data Science and Artificial Intelligence
8020 Graz, Austria
klaus.lichtenegger@fh-joanneum.at

Wolfgang Granigg
FH JOANNEUM – University of Applied Science,
Data Science and Artificial Intelligence & Global Strategic Management
8020 Graz, Austria
wolfgang.granigg@fh-joanneum.at

ABSTRACT

Vehicle Routing Problems (VRP) are of significant practical importance for many logistic applications, including disposal of waste. Many methods, mostly heuristics, have been proposed for (approximately) solving such problems. In this article, we explore a class of physically inspired approaches, which can be interpreted in an electrostatic or a thermal framework.

First, the approaches are been formulated in the continuum, where molecular dynamics methods are available. While the approach works in principle, it faces several problems (like orbits arising from angular momentum conservation) and thus is likely to be not very efficient. Therefore, we study a similar setup on a graph, where most of these problems do not arise in the first place. We compare results of agent-based simulations with results obtained with Simulated Annealing, as one of the standard heuristic approaches to the VRP.

While, in a first test, the well-established Simulated Annealing method performs better than the proposed method, the results are not far off, and in special situation (in particular for very large systems) the latter may actually be preferable.

CCS CONCEPTS

• **Applied computing** → *Physics*; **Transportation**; • **Computing methodologies** → **Agent / discrete models**; *Simulation evaluation.*

KEYWORDS

waste disposal, vehicle routing problem, optimization

ACM Reference Format:
Kerstin Lenger, Klaus Lichtenegger, and Wolfgang Granigg. 2022. Comparison of a Physically Inspired Agent-Based Model with a Simulated Annealing Algorithm for the Vehicle Routing Problem. In ,. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3543712.3543746

ICCTA 2022, 8th Int. Conf. on Computer Technology Applications

ACM ISBN 978-1-4503-9622-6/22/05…$TODO: PRICE
https://doi.org/10.1145/3543712.3543746

1 INTRODUCTION

1.1 The Vehicle Routing Problem

The Travelling Salesman Problem (TSP) is a famous problem of combinatorial optimization. Finding the optimal round trip connecting N cities, visiting each city exactly once, has a computational complexity of $O((N-1)!)$, which quickly defies brute-force approaches.

Heuristic approaches, including Simulated Annealing [9], Tabu search [3, 4] and genetic algorithms, however, have been successfully applied to this problem and typically yield satisfying (though not necessarily optimal) solutions with moderate computational effort [14]. The generalization of the TSP is the Vehicle Routing Problem (VRP) [1]. The VRP pursues the same goal but, with the difference that several routes are allowed to cover all cities. Moreover, each city has a certain demand of goods which must be fulfilled.

Problems of this type also arise in the field of waste disposal, and realistic setups have been studied by the authors in cooperation with pink robin gmbh, which offers a waste disposal service for construction sites on www.wastebox.biz. Due to various constraints like desired time windows, service times, capacity of the vehicles, dependency of possible disposal sites on the type of waste, special restrictions for dangerous wastes, this is an extremely complex problem.

For the current article, the problem is radically simplified to the task of picking up some goods at sources and delivering them to sinks (dumps), without further constraints.

1.2 Physical Analogies

In many cases, optimization problems can be solved by invoking analogies from physics, with Simulated Annealing and Simulated Tempering being particularly successful examples for this [2]. Often, magnetic analogies are used, and important models from computational intelligence has turned out to be equivalent to established physical models for magnetism. For example, Hopfield networks are equivalent to the Ising model for zero temperature [6, 10]. Also the VRP can be approached by using magnetic analogies [7, 8]. There are, however, physically inspired approaches from other fields that might be used for solving versions of the transport problem.

- *Electrostatic model*: In this picture, the sources are modelled by static positive charges, the sinks by static negative ones. Transport vehicles are treated as negatively charged when empty (thus being attracted by the sources and repelled by

the sinks) and positively charged when loaded (attracted by sinks and repelled by sources).
- *Thermal model*: Here, the sources are interpreted as heat sources (with source strength proportional to the remaining load) and the sinks as heat sinks. Transport vehicles are heat-seeking when empty and heat-avoiding when loaded. In contrast to the electrostatic model, there is no interaction between individual trucks.

Both models are to be supplemented with forces confining the vehicles to some finite space and can be enhanced with additional forces, as it will be discussed in Sec. 2.2.

1.3 Structure of the Article

After the introduction given in Sec. 1, we discuss the physical foundations in Sec. 2.1 and the continuous electrostatic problem in Sec. 2.2. The general translation to a network model is discussed in Sec. 3.1, the model implementation in Sec. 3.2. Simulation results and a comparison to a Simulated Annealing approach are given in Sec. 3.3. The article concludes with summary, conclusions and outlook, given in Sec. 5.

2 CONTINUOUS SETUP

2.1 Physical Foundations

Poisson's equation that governs the electrostatic potential Φ in the presence of a charge density ϱ_{el},

$$\Delta\Phi(\boldsymbol{x}) = -\frac{\varrho_{\text{el}}(\boldsymbol{x})}{\varepsilon} \tag{1}$$

is an elliptic partial differential equation (PDE) and formally equivalent to the static limit $t \to \infty$ of the heat equation for temperature T in the presence of fixed heat sources/sinks with density ϱ_{th}

$$\Delta T(\boldsymbol{x},\, t) - \frac{1}{a}\,\frac{\partial T(\boldsymbol{x},\, t)}{\partial t} = -\frac{\varrho_{\text{th}}(\boldsymbol{x})}{a\, c_{\text{vol}}}, \tag{2}$$

with ε, a and c_{vol} denoting material-dependent constants (electrostatic permittivity, thermal diffusivity and volumetric heat capacity). Δ denotes the Laplace operator, i.e. the rotationally invariant generalization of the second derivative. Assuming M fixed point charges Q_i at positions $\boldsymbol{p}_i$ and N mobile charges q_j at positions $\boldsymbol{x}_j = \boldsymbol{x}_j(t)$ yields a charge density

$$\varrho_{\text{el}}(\boldsymbol{x}, t) = \sum_{i=1}^{M} Q_i\, \delta\left(\boldsymbol{x} - \boldsymbol{p}_i\right) + \sum_{j=1}^{N} q_j\, \delta\left(\boldsymbol{x} - \boldsymbol{x}_j\right), \tag{3}$$

formally written that way using the Dirac delta functional δ. The resulting potential for $D = 3$ dimensions is given by

$$\Phi(\boldsymbol{x}, t) = \frac{1}{4\pi\,\varepsilon} \sum_{i=1}^{M} \frac{Q_i}{\|\boldsymbol{x} - \boldsymbol{p}_i\|} + \frac{1}{4\pi\,\varepsilon} \sum_{j=1}^{N} \frac{q_j}{\left\|\boldsymbol{x} - \boldsymbol{x}_j\right\|} \tag{4}$$

and the corresponding electrostatic force on the k^{th} vehicle is given by

$$\boldsymbol{F}_{\text{el}}^{(k)} = \frac{q_k}{4\pi\,\varepsilon} \left(\sum_{i=1}^{M} Q_i\, \frac{\boldsymbol{x}_k - \boldsymbol{p}_i}{\|\boldsymbol{x}_k - \boldsymbol{p}_i\|^3} + \sum_{j\neq k}^{N} q_j\, \frac{\boldsymbol{x}_k - \boldsymbol{x}_j}{\left\|\boldsymbol{x}_k - \boldsymbol{x}_j\right\|^3} \right) \tag{5}$$

In general, in D dimensions one finds a power law $\boldsymbol{F} \propto \left\|\boldsymbol{x}_k - \boldsymbol{x}_j\right\|^{-D} (\boldsymbol{x}_k - \boldsymbol{x}_j)$ for the force stemming from a single charge at position $\boldsymbol{x}_j$, corresponding to the geometric dilution.[1] Such a simple analytic form for the resulting force is not available for the time-dependent thermal setup described by (2). This formulation, however, is well-suited for treatment on a network, as discussed in Sec. 3.

2.2 Continuous Electrostatic Problem

The setup of a simulation essentially based on (5) is straightforward and can make use of well-established methods from classical molecular dynamics simulations, e.g. use of velocity-verlet or leap-frog algorithms [11], [13, Ch. 7].

In addition to the electric forces, boundary forces have to be introduced to confine the moving particles to a region close to the charges. The most obvious choice for this is a harmonic oscillator potential, which leads to linearly rising forces,

$$F_{\text{harm},\,i}(\boldsymbol{x}) = -k\,(x_i - x_{\text{right},\,i})\,\Theta(x_i - x_{\text{right},\,i}) - k\,(x_i - x_{\text{left},\,i})\,\Theta(x_{\text{left},\,i} - x_i), \tag{6}$$

written with the Heaviside step function Θ, a spring constant $k > 0$ and with $[x_{\text{left},\,i},\, x_{\text{right},\,i}]$ as the main interval for the coordinate x_i.

2.3 Extensions and Modification

The basic setup given by (5) and (6) can be extended or modified in several ways:

- Since the objective is not a realistic physical description of a system, the exponents of forces could be generalized. Also a cutoff distance could be introduced (which is common for rapidly decaying potentials like Lenard-Jones [11, Sec. 2.2], but is usually not applicable to the physical modelling of the long-range Coulomb interaction).
- An artificial increase of the "radiative power" of a positive charge that has not been visited for a while can help to attract negative charges. This becomes particularly relevant if a positive charge (source) is surrounded and thus screened by several static negative charges (sinks).
- Frictional forces can be introduced. A standard way is Newtonian friction

$$\boldsymbol{F}_{\text{fric}}(\boldsymbol{v}) = -\gamma\ \|\boldsymbol{v}\|\ \boldsymbol{v}, \tag{7}$$

  which can be modified to set in component-wise only above some threshold velocity v_{lim},

$$F_{\text{fric},\,i}(\boldsymbol{v}) = -\gamma\ (\|\boldsymbol{v}\| - v_{\text{lim}})^2\ \Theta(\|\boldsymbol{v}\| - v_{\text{lim}})\ \frac{v_i}{\|\boldsymbol{v}\|}\,. \tag{8}$$

- Small stochastic forces, modelling the effect of a heat bath, can by useful by eliminating situations, where a truck is stuck in an unstable equilibrium. By introducing such forces, however, determinism is lost, and a statistical analysis of several simulation runs is needed in order to make sound statements about the quality of the approach.

2.4 Preliminary Impressions

Simulation studies show that in general the straightforward electrostatic method is not very efficient in solving the VRP. In particular, the setup is often "too physical", i.e. angular momentum effects

[1] Note that this is obtained in an instantaneous approximation. In general, moving charges would require the treatment of retardation effects. This effect, however, is negligible for velocities far smaller than the speed of light.

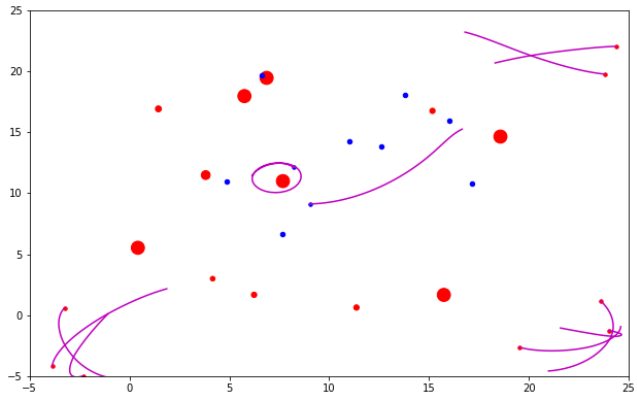

Figure 1: Screenshot of a typical simulation run, showing also one of the typical orbits that can form in the process. Sources are positively charged, sinks negatively, while trucks can be either. For the trucks, the trajectories of the last few timesteps are shown as well. For this simulation run, only for low friction forces and a moderate increase of radiative power, as described in Sec. 2.3 have been used. The size of the positive charges is enlarged proportional to this increase.

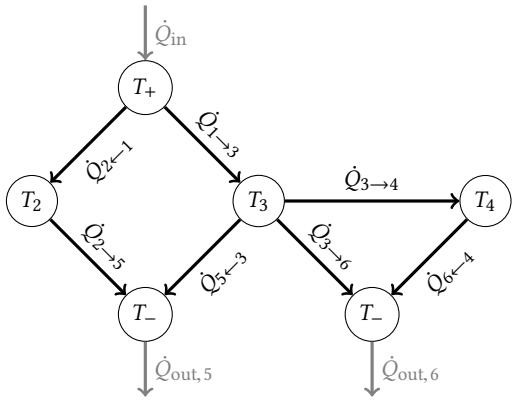

Figure 2: Heat flow in an exemplary network: Temperatures $T_1 = T_+$ and $T_5 = T_6 = T_-$ are held fixed (by implicitly assuming source and sink heat flows $\dot{Q}_{\text{in},i}$ and $\dot{Q}_{\text{out},i}$) while the other temperatures T_i are calculated from the thermal energy H_i according to eq. (10). The temperature differences result in heat flows $\dot{Q}_{i\to j} = \dot{Q}_{j\leftarrow i}$, determined by eq. (11), which in turn can change the energy content and thus the temperature of intermediate nodes.

create orbits, which in turn prevent a mobile charge from reaching a static attractive charge. Such a case is shown in Fig. 1.

This effect can be largely eliminated by sufficiently strong frictional forces like (7) or (8) for small values of v_{lim}). Too much friction, however, slows down the whole process; thus, some balance has to be found. The increase of radiative power, while often useful, can also lead to situations where the positive charges exert repulsive forces too strong to be overcome by attraction. In this case, the positively charged trucks are pushed into the boundary regions and are unable to ever reach the sinks an unload.

Hyperparameter tuning is expected to yield a viable, maybe even efficient method for the continuum. This line of research, however is still work in progress, thus we will not further elaborate on this in the current article. Instead we move to the – in general more relevant – setup on a discrete network, where most of these problems do not exist in the first place.

3 SETUP ON A DISCRETE NETWORK

3.1 General Considerations

The thermal version of the model is composed of nodes that act as heat sources, nodes that act as heat sinks, intermediate nodes and links, through which the heat flows according to the Fourier law. Each node is characterized by source and sink fluxes $\dot{Q}_{\text{in}}$ and $\dot{Q}_{\text{out}}$ and the thermal storage capacity C_i The thermal energy (very roughly the "heat", more precisely the enthalpy) H_i stored at a node i is governed by the differential equation

$$\frac{\mathrm{d}H_i}{\mathrm{d}t} = \dot{Q}_{\text{in},i} - \dot{Q}_{\text{out},i} - \sum_{j\in\Gamma(i)} \dot{Q}_{i\to j} \tag{9}$$

where $\Gamma(i)$ denotes the set of all neighbours of i.

The temperature at the node is given by

$$T_i = T(H_i) = T_0 + \frac{H_i}{C_i} \tag{10}$$

with $T(H=0) = T_0$, an arbitrary, but universal offset. Thermal flow through a link $i \to j$ with length $\ell_{i,j}$, connecting nodes with temperatures T_i and T_j, is given by

$$\dot{Q}_{i\to j} = -\frac{1}{R}\frac{\Delta T_{i,j}}{\ell_{i,j}} \qquad \text{with} \qquad \Delta T_{i,j} = T_i - T_j, \tag{11}$$

where R is a resistance parameter. In the thermal context one has $R = \frac{1}{\lambda A}$ with the thermal conductivity λ and the section area A (assumed to be constant).

If no node is permitted to act both as source and sink (as it is reasonable for application to the transport problem), the situation can by simplified by making input and output flows implicit, assuming all source nodes to have a constant temperature T_+ and all sink nodes a constant temperature T_- with $T_- < T_+$ (heat bath approach). All intermediate nodes i will than assume some temperature T_i which fulfills $T_- < T_i < T_+$. This is illustrated graphically in Fig. 2.

In the context of transport, the resistance R in eq. (11) can be interpreted as the reciprocal average traffic velocity $v_{i\to j}$,

$$\dot{Q}_{i\to j} = -v_{i\to j}\frac{\Delta T_{i,j}}{\ell_{i,j}} = -\frac{\ell_{i,j}}{t_{i\to j}}\frac{\Delta T_{i,j}}{\ell_{i,j}} = -\frac{\Delta T_{i,j}}{t_{i\to j}}. \tag{12}$$

Assuming symmetry, $t_{i\to j} = t_{j\to i} =: t_{i,j}$, the heat flow is characterized only by the temperature difference $\Delta T_{i,j}$ between the connected nodes and the time $t_{i,j}$ needed to travel the link or – assuming all travel velocities to be equal –, simply the link length $\ell_{i,j}$.

In this case, at node i, the heat-seeking particles move in the direction of steepest ascent, i.e. choose as next node j the one which

maximizes the directional temperature derivative

$$\frac{\Delta T}{\Delta \ell} = \frac{T_j - T_i}{\ell_{i,j}} \tag{13}$$

and heat-avoiding particles choose the node that minimizes it. At sources or sinks, the particles load or unload and consequently change their characteristics. Assuming a static situation, any ascent or descent route will end after a finite number of steps at a source or sink, since no local extrema can occur. If sources are removed by the loading process, temporary local maxima can form, but will be eventually eliminated again by the resulting heat flows.

When only considering the static limit, the system essentially is reduced to a set of (typically sparse) linear equations for the temperatures at the nodes. This system can be solved directly, but also – somehow better suited to an agent-based approach – iteratively by repeatedly re-determining node temperatures as weighted sum of the neighbour temperatures. As an example, for the simple system depicted in Fig. 2, one obtains a partially decoupled system of equations

$$\left(\frac{1}{\ell_{1,2}} + \frac{1}{\ell_{2,5}}\right) T_2 = \frac{T_+}{\ell_{1,2}} + \frac{T_-}{\ell_{2,5}} \tag{14}$$

$$\left(\frac{1}{\ell_{1,3}} + \frac{1}{\ell_{3,4}} + \frac{1}{\ell_{3,5}} + \frac{1}{\ell_{3,6}}\right) T_3 - \frac{1}{\ell_{3,4}} T_4 = \frac{T_+}{\ell_{1,3}} + \left(\frac{1}{\ell_{3,5}} + \frac{1}{\ell_{3,6}}\right) T_- \tag{15}$$

$$-\frac{T_3}{\ell_{3,4}} + \left(\frac{1}{\ell_{3,4}} + \frac{1}{\ell_{4,6}}\right) T_4 = \frac{T_-}{\ell_{4,6}} \,. \tag{16}$$

3.2 The Model

The model is implemented in NetLogo, a widely used agent-based software [15, 16]. The Interface of the model can be seen in Fig. 3. An agent-based Simulated Annealing algorithm was additionally implemented in NetLogo to be able to evaluate the performance and the results of the physical inspired model. To ensure a valid comparison between those two models, a separate file was created which generates the road network.

3.2.1 Initialization of the road network: The road network consist of undirected links representing the streets and "turtles" (agents), which represent the places. Further, a variable was introduced to define the number of links this node adds to the network. To make the network as realistic as possible, each link is assigned a `driving-duration` which acts as a weight. These driving durations are calculated with a normal distribution, where mean and standard deviation can be set manually, with non-positive values being rejected.[2]

In addition, trucks, dumps and containers, which are represented by turtles, were randomly distributed over the places in the network. Moreover, it is ensured that trucks and dumps are not assigned to the same place. The `nw` extension [15] has been used to save the created network with the dependencies between turtles and links.

3.2.2 Setup: In terms of a diffusion process in the previously created road network, places with containers and dumps and places without them are assigned a temperature. Each place with a container or a dump gets a constant temperature which can be adjusted.

[2] A more consistent approach would be to draw the durations from a gamma distribution, but for practical tests, this makes little difference for the resulting problems.

The other places obtain their temperature according to equation (13) and (14)-(16). The excerpt of the NetLogo code for the calculation of the temperatures is:

```
ask places with [ not any? dumps-here and not any? containers-here ][
    let actual-location one-of places-on self
    ask places with [ link-neighbor? actual-location ][
      set neighbor-temperature (temperature - [ temperature ]
      of actual-location) / [ driving-duration ] of street
      ([ who ] of one-of places-on self) ([ who ]  of actual-location)
    ]
    set temperature-temp sum [ neighbor-temperature ] of  places with
    [ link-neighbor? one-of places-on myself ]
    / sum [ reciprocal-driving-time ]  of places with
    [ link-neighbor? one-of places-on myself ]
  ]
ask places with [ not any? dumps-here and not any? containers-here ][
    set temperature temperature-temp
]
```

For each truck the attribute `loaded?` is set to `false` indicating that all trucks are empty at the beginning of the simulation. Additional lines of code are introduced for traceability and the debugging process.

3.2.3 Go Procedure: The goal is to navigate the trucks to the containers and dumps based on the temperature distribution in the network. The direction of the truck is determined by the temperatures in neighbouring places. If a container is assigned to the truck (i.e. if `loaded?` is `true`), the next node is the one with the minimum temperature. If, however, the truck is empty (i.e. `loaded?` is `false`), it will move towards the maximum temperature. The calculation of the respective temperatures, in the adjacent places is based on the formula (13), where $l_{i,j}$ describes the `driving-duration` of the links. This step is enabled by the following lines of code:

```
let actual-location one-of places-on self
move-to max-one-of places with [ link-neighbor? actual-location]
[ (temperature - [ temperature ] of actual-location) / [ driving-duration ]
  of link-with actual-location ]
]
```

After each time step, a new diffusion process is started to adapt the temperature. When a loaded truck arrives at a dump, the `loaded?` status is set to `false`. Similarly, when an unloaded truck arrives at a container place the `loaded?` status is set to `true`. As soon as all containers have been picked up and the vehicles are no longer loaded, the algorithm stops.

3.2.4 Simulated Annealing Algorithm. In the first step, a route with the starting point is initialized for each vehicle. Based on this information the container places and the corresponding dump places is added to the route of the nearest truck. In the `go` procedure, randomly chosen containers from randomly chosen trucks are exchanged to improve the solution. The temperature is reduced exponentially, $T \longrightarrow q\,T$ with a `cooling-rate` $q \in (0,\, 1)$.

3.2.5 Simulation Comparison. To enable a fair comparison, both models are loaded in the initialization model with the Level Space extension [5]. The respective parameters in the two separate models are set in the initial model and can therefore be passed on with the Level Space extension. This method ensures the centrally setting of all parameters, which provides the basis for application of Behaviour Space framework (integrated in NetLogo).

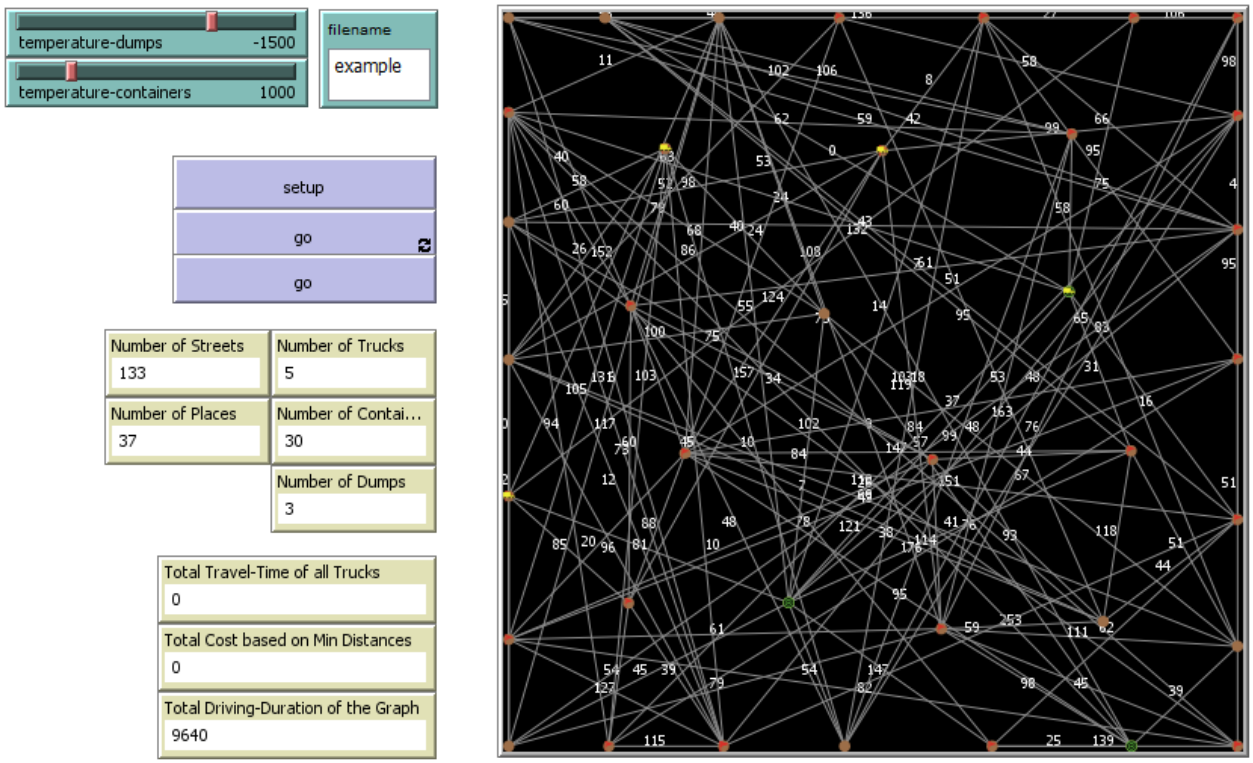

Figure 3: Interface of the NetLogo model

Table 1: Network parameter setting for the first simulation

`number-of-places`	155
`streets-outgoing-degree`	3
`driving-duration-of-streets-mean`	74
`driving-duration-of-streets-sd`	84
`density-of-dumps`	0.03
`number-of-containers`	50
`number-of-trucks`	5

3.3 Simulation Results and Comparison

For a sound comparison of the two models, two different simulations (settings I and II) have been carried out:

3.3.1 Simulation Setting I. The first setting includes 1000 files which were generated with the same network structure and similar dump places but with different container and truck places. The parameters for the problem setting are given in Tab. 1.

The simulation runs for the physically inspired model were carried out in advance and confirmed the expected independence of the results from the parameters `temperature-dump` and `temperature-container` (since those parameters only define the scale and can be changed by linear transformations without affecting the core algorithm). Thus fixed values for these parameters have been used in all subsequent runs. The model has been executed ten times for each base file (with different random locations for trucks and containers), which resulted in 10 000 runs in total. In order to have a sound basis for comparison, all combinations of the parameter values

$$\texttt{init-temperature} \in \{1\,000,\ 3\,000\}, \quad \texttt{cooling-rate} \in \{0.98,\ 0.99\}$$

have been tested for Simulated Annealing. As with the first model, for each parameter combination, ten simulation runs have been carried out, which resulted in 80 000 runs.

3.3.2 Simulation Results I. Of all parameter combinations tried, the best simulation results for the Simulated Annealing algorithm have been obtained with `init-temperature`= 3 000 and `cooling-rate`= 0.99, with `iterations`= 1 000. These results were used for the comparison. On average, the Simulated Annealing algorithm performed 15% better than the physically inspired model. In Fig. 4 the results are shown using a boxplot diagram.

3.3.3 Simulation Setting II. The second simulation setting deals with a fixed road network and locally fixed dumps. In both algorithms, containers and trucks are allocated randomly. To ensure comparability with the above results, 5 trucks and 50 containers were used in this setting. Furthermore, the best parameter setting from the Simulated Annealing algorithm was used. This was done 10 000 times for both algorithms to get a statement in regards to robustness.

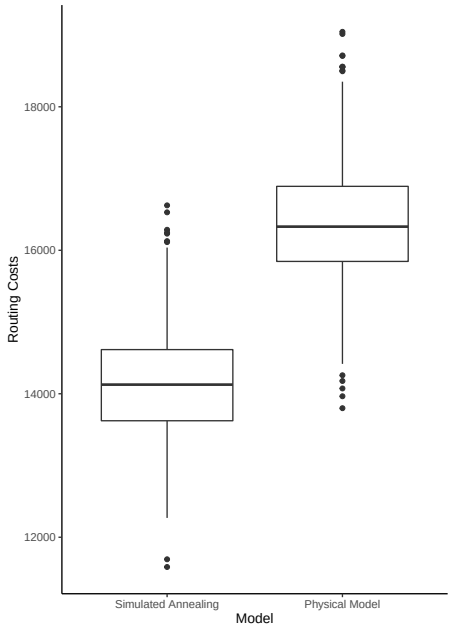

Figure 4: Comparison of the simulation results of setting I

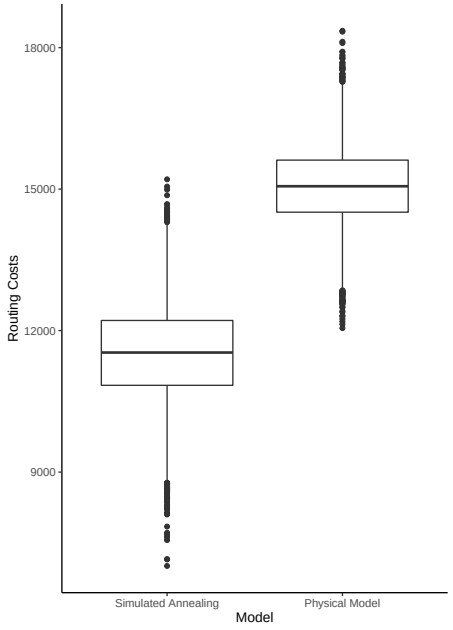

Figure 5: Comparison of the simulation results of setting II

3.3.4 Simulation Results II. On average, the costs of the physically inspired model were roughly 15 000. In contrast, Simulated Annealing performed significantly better and obtained an average cost of roughly 11 500, i.e. the results are roughly 30% better. This is also depicted in Fig. 5.

Comparing the execution times, the Simulated Annealing algorithm needs on average 0.4512 seconds and the physically inspired model needs 1.5614 seconds. An Intel(R) Core(TM) i7 − 97000K CPU (3.6 GHz, 32 GB RAM) with Windows 10 Version 1903, 64−Bit was used to perform the simulations.

4 COMPLEXITY AND SCALING PROPERTIES

A major advantage of the approach proposed in Sec. 3 is that it is strictly local and thus has favorable scaling properties. In order to show this, we consider a generic graph composed of v vertices and e edges (links), which contains m sources and m sinks. The average degree of the graph is given by $d_{\text{avg}} = \frac{2e}{v}$. On this graph, n_{tr} trucks move.

In each update step, $n_{\text{tr}}\, d_{\text{avg}}$ difference quotients have to be calculated on average to determine the movement of the trucks. The diffusion process, which simulates the heat flow, requires $n_{\text{rep}}\, v\, d_{\text{avg}}$ operations, with some small $n_{\text{rep}} \in \mathbb{N}$. Thus, the cost for one update step is

$$O((n_{\text{tr}} + n_{\text{rep}}\, v)\, d_{\text{avg}}) \equiv O(v\, d_{\text{avg}}) \equiv O(e), \tag{17}$$

since n_{rep} is a constant and typically we have $n_{\text{tr}} \ll v$.

Thus, the computational effort per step only scales linearly with the number of edges. This is in contrast to standard approaches like a greedy search, which would require to solve a shortest-distance problem, e.g. by Dijkstra's algorithm, see [12, Sec. 4.4], for all pairs of sources and sinks, which requires an initial effort of $O\left(m^2\,(e + v\,\log v)\right)$ and, in dynamic setups, additional $O\left(m\,(e + v\,\log v)\right)$ operations each time a source is depleted and replaced by a random new one.

For Simulated Annealing, studied as a benchmark in Sec. 3.3, the scaling properties are usually good, but typically a slower annealing process is required to obtain reliable results also for a larger number of trucks and of sources/sinks (while the number of edges is hardly relevant). Thus, the physically inspired method can be expected to be advantageous for setups on moderately sized graphs with a comparatively large number of sources/sinks and trucks.

5 SUMMARY, CONCLUSIONS AND OUTLOOK

We have implemented and tested a physically inspired method for solving a specific type of VRP. While the approach has performed somehow worse than using the well-established Simulated Annealing method, the results are not far off (15% and 30% longer routes for the specific setups).

In addition, only a rather static problem has been examined. In the case of a dynamic and/or stochastic formulation of the problem, with additional containers being introduced in the course of the simulation, the physically inspired model may gain better results than Simulated Annealing.

Also, as discussed in Sec. 4, for special setups (moderately sized graphs with a comparatively large number of sources/sinks and trucks) the physically inspired model may be advantageous. To obtain a deeper insights into such use cases, further investigations are required, including also comparisons with other approaches, e.g. from Ant-Colony optimization [17].

The same holds true for the continuous problem. While the electrostatic approach is not expected to outperform elaborate methods

of global optimization, it might be tuned to perform better than greedy methods, while at the same time, providing rather attractive scaling properties, at least if cut-off distances for the forces are introduced.

ACKNOWLEDGMENTS

The research has been performed in the *Big Data Analytics & Artificial Intelligence Research Center*, funded by the Federal Ministry for Science, Research and Economy within the COIN program, operated by the Austrian Research Promotion Agency (FFG) as project No. 866863, lead by Wilhelm Zugaj. The authors are grateful to Willibald Erhart, Manfred Füllsack, Georg Jäger, Raphaele Raab, Joachim Schauer, Gerhard Seuchter, Debora Stickler, Mario Wagner and Wilhelm Zugaj for valuable comments and discussions.

REFERENCES

[1] George B. Dantzig and John H. Ramser. 1959. The truck dispatching problem. *Management science* 6, 1 (1959), 80–91.
[2] Ai-min Deng, Chao Mao, and Zhou Yan-ting. 2009. Optimizing Research of an Improved Simulated Annealing Algorithm to Soft Time Windows Vehicle Routing Problem with Pick-up and Delivery. *Systems Engineering-Theory & Practice* 29, 5 (2009), 186–192.
[3] Fred Glover. 1986. Future paths for integer programming and links to artificial intelligence. *Computers operations research* 13, 5 (1986), 533–549.
[4] Fred Glover, Manuel Laguna, and Rafael Marti. 2007. Principles of tabu search. *Approximation algorithms and metaheuristics* 23 (2007), 1–12.
[5] Arthur Hjorth, Bryan Head, Corey Brady, and Uri Wilensky. 2020. LevelSpace: A NetLogo Extension for Multi-Level Agent-Based Modeling. *Journal of Artificial Societies and Social Simulation* 23 (01 2020). https://doi.org/10.18564/jasss.4130
[6] John J. Hopfield. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences* 79 (1982), 2554–2558. Issue 8. https://doi.org/10.1073/pnas.79.8.2554
[7] Mohd Muzafar Ismail, Muhammad Iqbal Zakaria, Zainal Abidin, Amar Faiz, Juwita Mad Juliani, Asrani Lit, Seyedali Mirjalili, Nur Anis Nordin, Mohamed Saaid, and Muhammad Faiz. 2012. Magnetic Optimization Algorithm Approach for Travelling Salesman Problem. https://www.academia.edu/24762539/Magnetic_Optimization_Algorithm_Approach_For_Travelling_Salesman_Problem.
[8] Biao-bin Jiang, Han-ming Chen, Li-na Ma, and Lei Deng. 2011. Time-dependent pheromones and electric-field model: a new ACO algorithm for dynamic traffic routing. *International Journal of Modelling Identification and Control* 12, 1 (2011), 29.
[9] Scott Kirkpatrick, C. Daniel Gelatt, and Mario P. Vecchi. 1983. Optimization by simulated annealing. *science* 220, 4598 (1983), 671–680.
[10] William A. Little. 1974. The Existence of Persistent States in the Brain. *Mathematical Biosciences* 19 (1974), 101–120. Issue 1–2. https://doi.org/10.1016/0025-5564(74)90031-5
[11] Dennis C. Rapaport. 2004. *The Art of Molecular Dynamics Simulation* (2 ed.). Cambridge University Press.
[12] Robert Sedgewick and Kevin Wayne. 2011. *Algorithms* (4 ed.). Addison-Wesley Professional.
[13] Benjamin A. Stickler and Ewald Schachinger. 2016. *Basic Concepts in Computational Physics*. Springer International Publishing. https://doi.org/10.1007/978-3-319-27265-8
[14] Paolo Toth and Daniele Vigo. 2014. *Vehicle routing: problems, methods, and applications*. SIAM.
[15] Uri Wilensky. 1999. NetLogo. http://ccl.northwestern.edu/netlogo/
[16] Uri Wilensky and William Rand. 2015. *An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo*. The MIT Press.
[17] Haitao Xu, Pan Pu, and Feng Duan. 2018. Dynamic Vehicle Routing Problems with Enhanced Ant Colony Optimization. *Discrete Dynamics in Nature and Society* (2018). https://doi.org/10.1155/2018/1295485