

An Approach to Model Network Dynamics of a Decentralized Supply Chain Network Using Optimal Predictive Control

Vladyslav Kuznetsov^{1,2,†,*}, Iurii Krak^{1,2,†}, Oleksandr Barmak^{3,†}, Hrygorii Kudin^{1,†}, Anatolii Kulyas^{1,†}, and Rostyslav Trokhymchuk^{2,†}

¹ Glushkov Institute of Cybernetics, 40, Glushkov ave., Kyiv, 03187, Ukraine

² Taras Shevchenko National University of Kyiv, 63/13, Volodymyrska str., Kyiv, 01601, Ukraine

³ Khmelnytsky National University, 11, Institutka str., Khmelnytsky, 29016, Ukraine

Abstract

The work discusses a novel approach to modelling supply chain networks that makes use of centralized approach on the supply centers and emergent behavior on the separate nodes of the network. This find its usage in the tasks of big data modelling and supply chain problems, which came of interest in recent time because of emergence of such area as e-commerce which is empowered by data driven technologies, such as data mining, intelligent data processing and AI. To address this task, we suggest: 1) approach to representation of network dynamics of separate supply chain nodes; 2) utilizing special methods and algorithms from theory of automated control, in particular model predictive control. In order to study this, we made an overview of the problem, suggested different levels of detail of our model, starting from the most simple ones, gradually approaching to more complex ones by engineering features of the models. We also addressed the issue of three-dimensional prediction horizon, that appears when modelling behavior of supply chains within spatial (node-wise) and temporal dimensions. To address this, we suggested to decouple the problem into separate dimensions, solving it accordingly by a set of methods, in particular, optimal model predictive control, radio frequency distribution problem, multiple travelling salesman problem and some others. The optimal model predicting control, devised in the paper, takes benefits from model predictive control method by adding an optimal control and constraining the system by a set of adjacent equations and Pontryagin maximum principle. As a result, we constructed few test models to study supply chain behavior in time, utilizing different models, like model predictive control, optimal predictive control and neural optimal differential controllers. As a result of conducted study, we got experimental results as well as analyzed an overall behavior of the model in terms of its stability, controllability and overall accuracy of the model. The experimental studies allow getting some useful recommendations to represent supply chain networks as well as advantages and disadvantages of different types of controllers, that define the behavior of nodes and the supply chain network in time.

Keywords

model predictive control, Pontryagin's maximum principle, decentralized big data systems

1. Introduction

The main goal of the study is focused into studying the underlying nature of decentralized supply chain networks and their behavior, including the supply chain analysis. There are a few problems to overview in this chapter of the paper, in particular:

1. *how the agents in the big data systems interact one with another (RQ1)?*
2. *how actions of each agent affect other agent and vice versa (RQ2)?*
3. *how the agents interact with the environment (RQ3)?*
4. *how one does describe the network structure and its behavior (RQ4)?*

14th International Scientific and Practical Conference from Programming UkrPROG'2024, May 14-15, 2024, Kyiv, Ukraine

* Corresponding author.

† These authors contributed equally.

✉ kuznetsow.wlad@mail.com (V. Kuznetsov); yuri.krak@gmail.com (Iu. Krak); alexander.barmak@gmail.com (A. Barmak); kudin@ukr.net (H. Kudin); anatoly016@gmail.com (A. Kulyas); trost@knu.ua (R. Trokhymchuk)

📄 0000-0002-1068-769X (V. Kuznetsov); 0000-0002-8043-0785 (Iu. Krak); 0000-0003-0739-9678 (A. Barmak); 0000-0002-1322-4551 (H. Kudin); 0000-0003-3715-1454 (A. Kulyas); 0000-0003-3516-9474 (R. Trokhymchuk)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

5. *Summarizing these questions, to address them, let's analyze them one by one.*

On RQ1, if analyzing the big data decentralized systems, one can observe, that the agents act in independent, or in other terms, emergent behavior, which makes actions of each actor likely unpredictable, if observed per each actor only. This makes creating policies, business decisions and strategies of stakeholders and owners of the supply chains as well as data brokers on such low level very uncertain which makes, on one hand the overall actions and responses quite unknown, adding a lot of risks in operating such systems, on one hand, and on another the problem of scalability of such systems in time and scalability of responses, making these systems to respond to user actions accordingly, depending on overall power of requests or density of these requests if you wish. To address this issue, one has to analyze the behavior of the system and agents of the system either on system level or in simple case whereas all interactions are isolated with no node-to-node interactions.

On RQ2, we have to admit, that despite the task of isolating informative features and observing them either on large or small scale, one may notice that all interactions between agents indeed are interconnected. This is presented in self-reinforcing effects, that include the network and synergistic effects [1-3]. The network effects describe how a network structure (nodes) can multiply actions of each other by multiplying the effect of the whole network over the sum of separate nodes. On the other hand, synergistic effects mean how actions of different actors within a system amplify actions of other actors, making, as a result, the overall power of interaction bigger than separate actions of actors, as if they acted independently. In overall, the network and synergistic effects may affect drastically the efficiency of the system, and they should be carefully considered and studied.

On RQ3, as a continuation of the RQ2, we must ask own self, what is important, the actions of separate agents or reaction of the environment. If these are the first, then the policy of the system is likely to be user driven, if the second, then it is likely system driven by policies, studied on overall behavior of agents and their interaction with the environment [4]. To do this, we can focus on approaches that are accepted and well known within the scientific community. One can classify them into separate categories, that study the system depending on 3 main criterions - first is how active the big data system compared to the agents of the system (criterion A), second is how active the agents in the system (criterion B) and how subjective factors - e.g. personal thoughts affect the system behavior (criterion C). By utilizing these criterions, one can define categories of big data systems models [5-8]:

- models based on physical and mathematical modelling (criterion A is bigger than B or C), in particular feedback loop, avalanche or key break effect models;
- models mostly economical by nature (criterion A and B are greater to C), which consist of Matthew effect models, game theory models and some other;
- models based on population models or models of spreading the diseases (criterion B is greater than A and C), for instance Frank's Bass model or any similar models;
- models that describe psychology or behavior of the large social groups (criterion C is greater than A or B), for instance Robert Gibrat's model or bandwagon effect.
- Summarizing these questions, to address them, let's analyze them one by one.
- After summarizing all of this, we can then step to RQ4, which asks us how the network structure is designed. For stakeholders or big business, the model is defined by their business centers and the dependent centers behavior is dictated by the policy or business strategy of the high tier centers. However, due to re-emergence of decentralized supply chain networks, one can assume that lower-level centers can make informed decisions based on the flow of information available on their level. It means that while the supply chain network can be centralized by design, it still can have features of decentralized networks, since the network defined rather by its behavior than by its design. Because of that, we want to focus our study mostly on the systems with hybrid behavior,

which in turn allows us to combine the advantages of centralized and decentralized approaches in one.

- Based on the answers to RQ1-RQ4, we can formulate the goals of the study, in particular in task of studying the model of the big data, which has a hybrid structure, to suggest models that describe behavior of such systems as well as to analyze model decision made by the suggested models in context of the task of optimization of flow of supply in supply chain networks with decentralized structure in time. To do this, we suggest solving such tasks:
 - to suggest a model of supply chain network, that describes the interaction of agents with environment and supply chain network;
 - to suggest a model of decentralized supply chain network, which utilizes a hybrid approach, using centralized structure with emergent behavior of separate nodes;
 - to compare the suggested model based on some already developed models in case of low level (one node) and system level (supply chain network)
 - to suggest an approach to simplify the model in terms of its complexity and time needed to find optimal model solutions and overall formulations

2. Methodology

To assess the result of the research, we have to suggest some methods, that may fit our purpose. Based on preliminary analysis (see RQ3), we can say, that models defined by the mathematical apparatus, and considering the system active over the agents are likely to be simpler to implement. However, we must take into account the simplicity vs accuracy tradeoff. Because of that we also must consider how the physical elements present in the decentralized network such as supply stores, distribution centers etc., and virtual elements like data centers and analysis software interact one with another. If classifying the models by their features, we can say that physical models somehow may lack features as the economic, social or human oriented models for instance [9], but in overall allow to narrow the simulation environment by the system itself. On the other hand, the economical models are more balanced, in terms of representing the users as well as the big data system as active elements. The population models are shifted more towards users or agents and consider them the only active elements, so as the system reactions are derived only as a consequence of independent actions of separate users. The social and psychological models increase this to a bigger extent, focusing on a will and desires of the users, rather than a “game” played versus the environment, which is the big data system in our case.

To assess this problem, we devise to focus mostly on the physical or mathematical models, so as we can develop them more easily and have a bigger control over the environment by integrating strategies to counter synergistic effects in the model by countering them using adaptive control of the decentralized supply network. To do this, we have to assess the amount of disruption or disturbance created by agents and to compensate it via means of the models we do consider. In this case we can observe this as sort of feedback loop [10], whereas the agents make some actions and the system responds to their action, feeding the input to compensate the difference, using from the automated control theory thereof.

3. Identifying the Parameters of the Model

To create a model, at first, we must define the set of parameters within a decentralized supply network. To do this, we may analyze the interactions within a system. This can be explained by a simplified model, which connects the utility and usefulness of the system with actions of the agents in the system. The agents create some interactions that generate feedback, communication, engagement and different type of interaction of the system.

Based on these general considerations, we can assume that there is a set of controlled as well as non-controllable parameters within a system. As a result, a parameter set can be defined as a following tuple:

$$P = \langle F, E, T, S, V \rangle. \quad (1)$$

Whereas P is a set over parameters F, E, T, S, V which define own type of action or interaction; F defines the feedback provided by agents, and basically it is a positive feedback value, which is passed to the input of the system, which contributes to new actions of users and control the amount of the input disruptions; E, on its hand is a level of engagement or interest of the agents that define the likelihood of the interaction, made by their own, in simple words it can be used as a measure of usefulness or utility of the system overall; T is an amount of trust the agents put into system decisions and overall trustworthiness of the algorithms that drive the system; S is an amount of user satisfaction, which defines how well the system is answering their needs; V is for value concept from 4V concept (value, volume, velocity and veracity) from big data system which allows accounting for data value itself, basically similar to trustworthiness, but pointed to the users instead [11].

Obviously not all of these parameters can be controlled or observed, by the system, the P set or tuple defines the qualitative parameters of the system, when the quantitative parameters are ones that can be easily measured and monitored via the system [12].

These parameters can be separated into two big categories, which include user and system-controlled factors, as well as derived factors (integral or differential characteristics): demand which shows how often the users interact with the system, connected with their specified needs for goods or services depending on the platform; supply, which shows how good the system can satisfy the needs of the users and can be considered literally the opposite of what the demand is.

To add this all above, we can now introduce some other parameters as: volume or stocks, available at the warehouse as a reserve; resupply speed, which shows the highest momentary demand to supply in time; transaction quantity, which defines the overall number of interactions per user; transaction speed, which is the same as above, but divided by average time; ordering quantities - which is similar to demand, but is a cumulative sum of them; utility – the overall efficiency of the system.

These parameters are very important, but in real time situations, one have to consider ones that can be utilized in some formulae or optimization procedures in real time scenarios [13].

4. A Model of Action and Reaction in a Potential Field of Forces

To start a task, let's consider a model of a node in a supply chain, that is affected by two actions - the action of the environment and the response of the system to such actions. To assess this, we may utilize some known models, that are used both in robotics and to describe actions in some kinematic systems. For, instance, here we suggest applying the approach to formalize interactions in manipulation systems, using a model of action and reaction in a potential field of forces. This model is quite known from theory of automated control as well as robotics to plan the movement of a manipulator system [14, 15].

We can utilize this approach by defining our key parameters, elements, as well as interactions: the network in the simplest one-chain case represents a motion planning system whereas the central node allows interpreting the central warehouse, whereas all child nodes represent the distribution centers of the low level. Approaching our task with such analogy, we can benefit from known model by utilizing it in the other area of research.

Let's define a general model first. Consider a model of a manipulator system (a node), that acts within a potential field of attractive and repulsive forces; to describe the model let's introduce the potential field U , as well as U_{att} and U_{rep} that represent the attractive and repulsive potentials respectively. In this case, U can be defined as follows:

$$U = U_{att} + U_{rep}, \quad (2)$$

whereas

$$U = \frac{1}{2} k_{att} \|goal - position\|^2 \quad (3)$$

$$U = \frac{1}{2} k_{rep} \sum_i \|position - obstacle\|^2 \quad (4)$$

where $goal$ defines a target position of the manipulation system, $position$ is a current coordinate and $obstacle$ is a certain obstacle or place to avoid or to overcome.

We can now narrow the general model into a narrow, partial case, where we consider actions in a decentralized supply chain network, in its node; to do this we can substitute some terms in previous equation, in particular replacing attractive and repulsive potentials with potentials of demand and supply – U_d and U_s respectively. This overall allows us to define the relationship between supply and demand by the modified model of potential fields:

$$U = U_d + U_s, \quad (5)$$

Since we operate slightly different values than physical attractive and repulsive forces, but quantitative characteristics of the system, such as supply and demand, we must modify the equations for the substituted values – U_d and U_s :

$$U_d = \frac{1}{2} k_d \|G - e\|^2, \quad (6)$$

$$U_s = \frac{1}{2} k_s \sum_i \|e - \delta p_i\|^2 \quad (7)$$

where the G term defines the goal of the system, that is represented by a desired amount of demand, e , on the other hand represents an epsilon value, which shows the amount of input disturbance on demand, and δp_i define the policy that restricts or suppresses some amount of demand to exclude the situation of infinite growth of supply. This substitution allows us to utilize the same principles that are used in a single chain in a manipulator system in the supply chain that is also represented by a single chain of sequential nodes.

As a result, we can design a supply chain network as a composition of separate chains, that act simultaneously; because of that on each node as well as a separate chain we can define a separate controller, a chain of controllers as well as a network of controllers. Thus, in such a network, the controllers generate reactions on a specific event which is represented by the input demand disturbance and this reaction is measured as a change of supply on a fixed time step. This approach allows us not only to construct a model of a complex supply chain network, but also allowing to have independent network dynamics within a supply chain.

5. A Model of a Node in a Supply Chain Using Model Predictive Control

To assess the task of modelling the supply chains, we devise starting from the simplest models (one node) to more complex (a chain and or a network). In the case of single node system, one has to select a model which is best suited to fulfill such a task. Based on our overview done in the section

Introduction, we may say that quite many models may fulfill such a task, for instance economical models, models from game theory, social models and physical models. Among these models, we considered that physical ones are more fitting our task, in particular models from theory of automatic control and robotics.

Our model, suggested before, such as model of attractive and repulsive forces is quite promising, however, we think we can modify it or substitute by more advanced models, that take in account the reaction of the system and its feedback, which underscores the models based on a feedback loop. The benefit of this model that they introduce positive or negative control into input, depending on the value of the input disturbance and prediction of its changes based on the previous observations. This introduces time dynamics and allow accounting for changes in time, which is very important in modelling of such systems.

To suggest a better analogy, we may consider 4 types of regulators, that are commonly used to control processes in dynamical systems of different nature, and based on theory of optimal control as well as robotics and theory of automatic control in general. These are neural regression model, proportional integral and differential controllers (PID), model predictive control (MPC) as well as neural optimal differential controllers (NDEC) [16].

These models differ by their complexity as well as accuracy of their predictions in time. In general, the neural regression is a simple analytical function or regression constructed on the data, which consists of previous observations of the system in a fixed amount of time. While these models are simplest, they tend to overfit on data and work the best on typical situations, which may happen quite unlikely in big data systems, that have non-deterministic nature and unpredictable behavior of agents in the system. The PID regulators make some step forward, since they account for integral and differential parameters of the system, which allows accounting to some changes on short interval of time and being adaptive to typical trends such as growth or decline of the input parameters. However, they are not as accurate as MPC regulators, for instance. MPC regulator, on its hand includes as a partial solution a simple regulator, however, it defines an observation horizon, over which the control is performed as well as prediction horizon over which one can predict the change of parameters in time [17, 18]. It also includes optimization procedures (which will be discussed further), that enable near-optimal control of the variables in time. There are more advanced models, like neural-optimal differential controllers that combine features of neural regulators and model predictive control models in one: each step of prediction horizon is approximated by the neural regression, and its optimal solution is found by following an optimization procedure, in similar way, as done in MPC regulator, however these solutions are approximated with neural regressions.

Because some models didn't answer our requirements in terms of their stability and overall ease of implementation, we considered the MPC model as a prototype of the model of a node of a big data system, assuming that it has only one input and one output. In this case the model allows representing, a chain in a supply chain network separately and model spatial-temporal dynamics per node separately if needed.

Let show a simple solution. The task is to find the optimal (minimal) solution for cost function J, based on overall costs that are comprised of ordering, delivery and storage on the fixed and defined prediction horizon N. Consider the objective function J:

$$J = \sum_{k=0}^{N-1} (U_k^i C_o + Y_k^i C_h) \quad (8)$$

whereas the terms U_k and Y_k are column vectors of orders and available stocks (volumes of goods or services) on a given time step l , the cost vectors are defined as C_o and C_h as cost per order and storage of a single unit respectively.

The objective function is bound by system dynamics constraint, which is denoted as follows

$$X_{k+1} = X_k + U_d - D_k, \quad (9)$$

where X_{k+1} defines the state (state value) on a following step $k+1$, X_k – on a current step and U_d is utility volume, and D_k demand value on k -th step respectively.

The supply is constrained by the expression

$$Y_k = X_k \quad (10)$$

which denotes that the volume of goods, available at the moment, at any given k th step is equal to the state variable on the same time step.

The productivity is bound by inequality:

$$U_{min,k} \leq U_k \leq U_{max,k}, \quad (11)$$

where U_k , $U_{min,k}$, $U_{max,k}$ are current, minimum and maximum ordering quantities on a k -th time step respectively.

Based on that we assume that on initial stage

$$X_0 = x, \quad (12)$$

where x is the initial state of the studied system.

Doing so, we may utilize different approaches to get an optimal solution using an optimization procedure presented above. The key outcome is that the optimization procedure tries to minimize the running cost by ensuring that the solution satisfies the dynamics, supply and productivity constraints within the desired prediction horizon [19].

We may modify this model further, by introducing the Pontryagin maximum principle [20]. The key idea lies behind is to introduce cost or costate variables, that complement a system of equations per each constraint and state equation. These dynamics constraints allow one to ensure the optimal (we speak about minimal in terms of cost) solution of the system. Let's modify the previous example (7) to assess the Hamiltonian:

$$H = \sum_{k=0}^{N-1} (U_k^i C_o + Y_k^i C_h + \lambda_k^i (X_{k+1} - X_k - U_k + D_k)) \quad (13)$$

whereas, λ_k is the cost variable (costate) bound to the state equation at a given time step k . The dynamics constraint bound with the costate value allows one to utilize the costate multipliers, which affect the general solution.

$$\frac{d\lambda_k}{dt} = - \frac{\partial H}{\partial X_{k+1}} \quad (14)$$

In order to utilize the maximum principle, the Hamiltonian has to be bound by conjugate equations, also called as costate equations [20]. These equations can be obtained by applying partial derivatives of the Hamiltonian with respect to each state variable. To derive the Hamiltonian, let's substitute the H value first:

$$\frac{d\lambda_k}{dt} = - \frac{\partial}{\partial X_{k+1}} (U_k^i C_o + Y_k^i C_h + \lambda_k^i (X_{k+1} - X_k - U_k + D_k)) \quad (15)$$

Since the partial derivative is taken with respect to a specific value of k , all other terms of the sum (i.e. on $k-1$ and $k+1$ step) will be zero, since derivative of a constant is always zero.

To get a specific form of conjugate equations, let's unravel the expression. Since other terms but X_{k+1} , such as $U_k C_o$ and $Y_k C_h$ don't depend on itself, we can simplify the expression.

By expanding the expression further, we get a linear ordinary differential equation (ODE) which for specific k can be written as follows in a general form (using substitute for $d\lambda_k / dt$ as λ_k'):

$$\lambda_k' + \lambda_k = 0 \quad (16)$$

If we consider a system with N states, we also get a set of N conjugate linear ODEs:

$$\begin{aligned} \lambda_1' + \lambda_1 &= 0 \\ \lambda_2' + \lambda_2 &= 0 \\ &\dots \\ \lambda_N' + \lambda_N &= 0 \end{aligned} \quad (17)$$

So, the resulting set of conjugate ODEs allows utilizing the principle of optimal control by having the conjugate equations per each given time step k in prediction horizon N . In practical applications, however, this system of conjugate equations has to be solved first, and only after that being applied directly to optimization problem, for the control variables U_k to define the policy that answer the requirements for optimal control depicted above.

6. A Model of a Supply Chain with Multiple Separate Chains

Usually models of supply chains are more complicated. In fact, we can observe a model having a network structure, that moves in time along the time prediction horizon. In this case one has to model complex time-spatial relationships which may not be practical for real time situations. So, to overcome this one may suggest per se some simplifications that include:

- simplification of the network structure overall;
- using some techniques to modify observation horizon;
- to propose a simplified model of the existing model.

Let's observe all these three possibilities. Obviously, some network structures can be combined into bigger units and hence allow one to deal with less amount of nodes and different branches of the tree. Ideally this has to be a binary tree that would allow easy traversal up and down and propagate solutions using exact same formulae on each level of the structure. This may have some positive things compared to the irregular structure that has the different number of child nodes in the different level so as the solution per each level and per each node would be quite unique, however still be applicable, so as one propagates the solution up by gathering the solutions on the lower levels using some techniques of combining solutions in one, that we will discuss a little bit later.

However, this approach has one serious flaw. The network on each level has only 1 level or prediction horizon in space and, contrary to example we discussed in the previous chapter, would likely have more than one step in time, which allow us to build a prediction function on a reliable number of observations and on quite long prediction horizon in the future steps. Also, we have to do something with the prediction horizon, which would have to possess a balanced number of points of prediction horizon in time. Since we observe the model in three dimensions, in means that the prediction horizon changes both in depth, width of the supply chain network and time as well. So, the next problem is to deal with the increased number of inequalities and conjugate equations if we suggest using Pontryagin maximum principle.

This give one simple solution - is to simplify the number of equations, tied to the system and somehow observe the network dynamics in time in such way, so we could solve as little number of optimization procedures as possible. The best way to do this is to modify the network, whereas we observe not a tree-like structure, but a set of independent parallel nodes of the same length which allows us to separate each coordinate in the prediction horizon and in fact, to decouple the system into a sequence of separate coordinates, in which we solve the system step by step per each coordinate of the prediction horizon.

So, in simple words the solution can be found as follows: at first, we solve the network dynamics in depth per each link in parallel chains, then combine the solutions to get solution in width of the network, and lastly solve the prediction horizon problem in time.

7. Finding an Approximate Solution for a Three-Dimensional Horizon

The main feature of the proposed simplified model is that it unravels a graph (network) from a hierarchical structure to another graph-like structure consisting of several parallel links of the same length. Despite the fact that they are independent, there is still a need to control the distribution of demand between distribution centers of goods or services and to reward their use in proportion to their size [21, 22].

The first option is to physically determine the location of distribution centers and weights of the network by solving the multiple traveling salesman problem (MTSP).

The problem can be formulated as follows: it is necessary to minimize the total distance traveled by each of the suppliers, taking into account a set of constraints:

$$\min \sum_{i=1}^P \sum_{j=1}^P l_{ij} \cdot x_{ij}, \quad (18)$$

where x_{ij} : a binary variable indicating whether the arc (i, j) is included in the route, l_{ij} : the distance (or path cost) between i and j , P : the number of nodes, M : suppliers.

At the same time, the following restrictions must be ensured. First, each node must be visited exactly once by exactly one supplier:

$$\sum_{i=1}^P x_{ij} = 1, j = 1, 2, \dots, P \quad (19)$$

$$\sum_{j=1}^P x_{ij} = 1, i = 1, 2, \dots, P$$

Restrictions for which each provider has an associated route:

$$\sum_{i \in S} \sum_{j \in S} x_{ij} = |S| - 1, S \subseteq \{2, 3, \dots, P\}, \quad (20)$$

$$2 \leq |S| \leq P$$

Binary constraints are also imposed on the decision variables:

$$x_{ij} \in \{0, 1\}, \quad (21)$$

where, $x_{ij} = 1$ indicates that node j is visited immediately after node i in the tour.

The second option is to use a joint solution of the problem of radio frequency distribution, which consists in finding the solution for the best distribution of the bandwidth and power of the transmitted signal. In the optimization procedure, a fixed amount of power, a non-zero transmitted power of each of the data transmission channels, and a non-zero bandwidth of the channel are used as constraints. This procedure can be modified by replacing the transmission rate with traffic and the capacity with reserves. If these parameters are replaced, the optimization procedure will consist of maximizing the expression with the constraints imposed on it:

$$\sum_{i=1}^P KL(\alpha_i \cdot \text{traffic}_i), \quad (22)$$

$$\alpha_i \cdot (\text{traffic}_i + \beta_i \cdot \text{supply}_i) - \alpha_i \cdot \beta_i \cdot \text{supply}_i$$

$$\text{supply}_i \geq 0, \quad (23)$$

$$\text{traffic}_i \geq 0, \quad (24)$$

$$\sum_{i=1}^P \text{supply}_i = \text{supply}_{tot} \quad (25)$$

$$\sum_{i=1}^P \text{traffic}_i = \text{traffic}_{tot} \quad (26)$$

where (22) is an optimization procedure, (23) is a supply constraint, (24) is a traffic constraint between distribution centers, (25) is a fixed amount of traffic, (26) is a fixed amount of supply, and

KL is the Kullback-Leibler's divergence , supply_i : supply at point i , where i =1,2,..., p, traffici : traffic (restocking rate) at point i, α_i and β_i : parameters related to endpoint i, supply_{tot}, traffictot – total permitted volume of supply and permitted traffic, respectively.

This way of solving the problem allows scaling it to other coordinates of the prediction horizon, since the prediction horizon must take into account 3 coordinate axes, where one axis is the depth of the supply chain, the second is the width of the supply chain, and the third is time. To overcome this, a solution is proposed, which involves the aggregation of individual network solutions on parallel links of the network structure, with the further restriction of the dynamics equations for the defined structure of the supply chain and the solution of the above problems.

In theory, this allows replacing an entire network of nodes or links with a single node that contains a solution that extends the network structure to a single element. This means that solutions are first propagated in the supply chain and then a partial solution is used for the initial conditions of the single-node structure, which allows us to focus on the time dynamics, with a loss of some accuracy. Thus, instead of a network with variable spatio-temporal dynamics, it is proposed to use a so-called "frozen" network, which is based on a partial solution for the prediction horizon.

8. Experimental Implementation and Testing

For a practical assessment of the proposed approach to solving the problem, a software implementation was developed that performs a step-by-step solution of the following problems: the problem of several traveling salesmen (to obtain the structure of the supply chain), the problem of forecasting in the depth of the chain supply with parallel branches (to obtain network dynamics), aggregation of results by a modified model of radio frequency distribution (to extend the solution and obtain initial conditions for a problem with one node) and solution of system dynamics in time using by the NDEC regulator.

The implementation used a number of libraries in the Python language, which included, in particular, cvxpy [23] - for describing the prediction horizon and network dynamics of individual chains, aggregation and prediction of the results of individual chains, nnc [24] for describing neuro-differential equations for one network node with one group of goods or services. To run, the test system had the OS Ubuntu 22 with machine learning libraries installed, including the above.

For the solution, data from open sources were used, namely, the structure of the branch of the Agromat company and the location of its warehouses and branches according to Google Ukraine map data.

According to the previous solution of MTSP, the supply chain forms a circular route in which there are two branches of approximately the same size (see fig. 1).

The main element is point 0 (the central composition), from which 2 main rays emerge, consisting of segments 0-2-3-5 and 0-1-4-6, respectively. This allows us to reduce this problem to the solution of a linear horizon with a depth of 3.

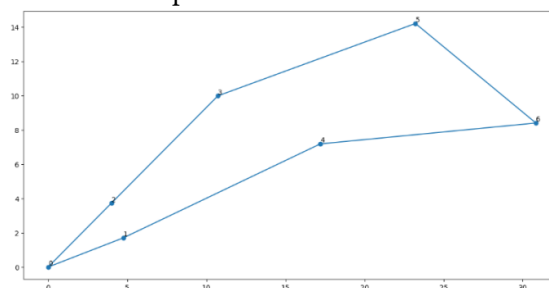


Figure 1: Structure of supply chain obtained from MTSP.

Let's consider the result of an experiment on modeling a model with a consolidated structure with one node and two types of regulators - MPC and NDEC. Below in fig. 2 shows the simulation plots for MPC.

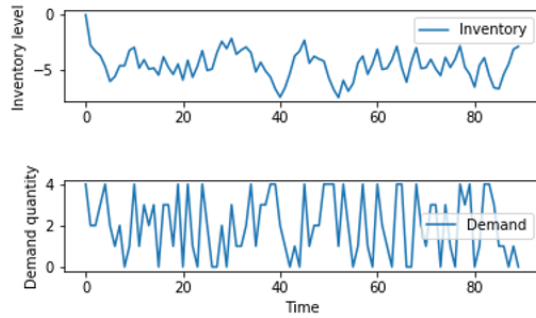


Figure 2: Simulation results for MPC.

During the experiments, the effectiveness of two distinct approaches to optimal control was compared. The first approach utilized the `cvxpy` library [23] to simulate the Model Predictive Control (MPC) regulator. The second approach employed the `nnc` library [24] to simulate the neuro-differential optimal controller (NDEC). Each approach brought unique methodologies and computational requirements to the table. The NDEC, in particular, involved solving the differential equations that govern system dynamics, which added a layer of complexity to the simulation process.

The simulation period for both methods spanned 100 virtual days. Within this period, supply and demand values were updated 10 times a day to closely mimic real-world fluctuations and dynamics. The primary objective of the optimization method was to minimize the costs associated with the supply of goods while maintaining an adequate stock level of goods or services to meet demand.

During the testing phase, it was observed that the implementation of the `nnc` library was highly demanding in terms of hardware and memory resources. Specifically, the `nnc` repository required a CUDA-compatible graphics card with at least 8GB of memory to effectively perform the computations. However, at the time of the experiments, the training of both the NDEC models and the MPC regulator was conducted on the central processing unit (CPU). This reliance on the CPU for calculations significantly neutralized the primary advantage of the NDEC models, which is their high-speed computational capability when run on a graphics processing unit (GPU).

Consequently, the execution times for training the MPC and NDEC models showed a stark contrast. The training of the MPC regulator took approximately 10 seconds, demonstrating its efficiency and speed. In contrast, a complete training cycle for the NDEC model took around 15 to 18 minutes, depending on the system load and computational intensity. This significant time difference highlighted the impact of not utilizing the GPU for NDEC training.

The training regimen for the NDEC models included 4000 epochs, within which the controller achieved an acceptable error margin after about 750 epochs. The asymptotic behavior of the training continued to improve up to the 4000 epochs mark, indicating that prolonged training did yield better results but at a considerable computational cost.

One notable advantage of the NDEC regulator was its adaptability and learning efficiency. Unlike the MPC regulator, which required constant re-optimization, the NDEC regulator needed retraining only at specific intervals. This intermittent training requirement reduced the computational burden and allowed for periodic updates to the control strategy, thus improving overall system performance.

The improved adaptability of the NDEC regulator was clearly demonstrated in the results. As illustrated in Figure 2 (right), the supply curve for the NDEC regulator closely followed the demand curve, indicating a more responsive and efficient control mechanism. Despite this advantage, the slow training speed on the CPU necessitated further investigation into methods to enhance efficiency. Future studies were planned to explore ways to leverage GPU capabilities fully and to optimize the training process, thereby reducing the time required for NDEC model training while maintaining its high adaptability and control precision.

In summary, while the MPC regulator offered speed and ease of use, the NDEC regulator provided superior adaptability and closer alignment with demand patterns. The primary challenge for the NDEC approach was the computational time required for training on the CPU, suggesting a clear need for hardware optimization and potential GPU utilization to unlock its full potential.

9. Conclusions

In this work, we proposed a comprehensive model, which is based on a predictive control model that effectively combines the features of a decentralized and centralized big data network. By utilizing this approach, specific models address the emergent (independent) behavior of each individual network nodes while keeping a centralized structure for overall ease of control. Using this approach, we integrate decentralized network dynamics approaches into a centralized framework, and obtain a robust approach to manage complex supply chain networks.

During the study, we formulated a problem of multidimensional prediction horizon. To address this problem, our model suggests an approach to obtain separate solutions for each coordinate of a three-dimensional prediction horizon by decoupling the system into separate dimensions such as width, depth, and time. This multidimensional approach improves our confidence of the solution and enables more precise and effective control strategies.

One of important results emphasizes on exploration of possibilities of an optimal model of predictive control, which achieves the requirements of optimal control by applying the Pontryagin maximum principle. To apply this approach, we also aggregated model solutions from different levels of the supply chain model (warehouses, distribution centers), in particular by utilizing multiple travelling salesman problem as well as power and frequency distribution problem, which allowed us to more easily understand and more easily implement the control mechanism within such synthetic networks.

In the experimental part of the study, we focused our tests on practical evaluation of the software implementation, by utilizing both Model Predictive Control (MPC) and neuro-differential optimal controllers (NDEC) to simulate supply-demand fluctuations within the system in space and time. The experimental results highlight some bottlenecks and potential issues that need to be evaluated in future research. For instance, the learning time of the NDEC controller was relatively big, which may be caused by some hardware limitations and memory bottlenecks that lead to the computational intensity and possible less efficiency utilization of the graphic processing unit (GPU) on the training stage. However, such a bottleneck does not affect the quality and accuracy metrics of the NDEC controller solution, but only highlights potential issues that need to be addressed in the future.

References

- [1] S. Mansouri, F. Castronovo, R. Akhavian, Analysis of the synergistic effect of data analytics and technology trends in the AEC/FM industry, *J. Constr. Eng. Manag.* 146.3 (2020) 04019113. doi:10.1061/(asce)co.1943-7862.0001759.
- [2] D. M. Haftor, R. Costa Climent, J. E. Lundström, How machine learning activates data network effects in business models: Theory advancement through an industrial case of promoting ecological sustainability, *J. Bus. Res.* 131 (2021) 196–205. doi:10.1016/j.jbusres.2021.04.015.
- [3] R. Alikhani, A. Ranjbar, A. Jamali, S. A. Torabi, C. W. Zobel, Towards increasing synergistic effects of resilience strategies in supply chain network design, *Omega* (2022) 102819. doi:10.1016/j.omega.2022.102819.
- [4] K. Berwind, A. Voronov, M. Schneider, M. Bornschlegel, F. Engel, M. Kaufmann, M. Hemmje, Big data reference model, In: *Big data analytics*, Chapman and Hall/CRC, Boca Raton: CRC Press, 2017, pp. 55-74. doi:10.1201/b21822-3.
- [5] S. Agrawal, S. Yin, A. Zeevi, Dynamic pricing and learning under the bass model, In: *EC '21: the 22nd ACM conference on economics and computation*, ACM, New York, NY, USA, 2021. doi:10.1145/3465456.3467546.
- [6] F. Arendt, Media stereotypes, prejudice, and preference-based reinforcement: toward the dynamic of self-reinforcing effects by integrating audience selectivity, *J. Commun.* (2023). doi:10.1093/joc/jqad019.
- [7] R. Farys, T. Wolbring, Matthew effects in science and the serial diffusion of ideas: Testing old ideas with new methods, *Quant. Sci. Stud.* 2.2 (2021) 505–526. doi:10.1162/qss_a_00129.

- [8] Y.-S. Hu, The impact of increasing returns on knowledge and big data: from adam smith and allyn young to the age of machine learning and digital platforms, *SSRN Electron. J.* (2019). doi:10.2139/ssrn.3414339.
- [9] Iu.G. Kryvonos, Iu. V. Krak, O.V. Barmak, A.S. Ternov, V.O. Kuznetsov. Information technology for facial expression analysis. *Cybernetics and System Analysis*, 51(1) (2015): 25-33. DOI:10.1007/s10559-015-9693-1
- [10] E. Katsamakas, O. V. Pavlov, Artificial intelligence feedback loops in mobile platform business models, *Int. J. Wirel. Inf. Netw.* (2022). doi:10.1007/s10776-022-00556-9.
- [11] H. Kasim, T. Hung, X. Li, Data value chain as a service framework: for enabling data handling, data security and data analysis in the cloud, In: 2012 IEEE 18th international conference on parallel and distributed systems (ICPADS), IEEE, 2012. doi:10.1109/icpads.2012.131.
- [12] F. Cicirelli, L. Nigro, Control aspects in multiagent systems, In: *Studies in big data*, Springer International Publishing, Cham, 2015, c. 27–50. doi:10.1007/978-3-319-23742-8_2.
- [13] Á. Bányai, B. Illés, Z. Gaziza, T. Bányai, P. Tamás, Impact of logistic processes on economic order quantity with quantity discount: an optimization approach, *Asian J. Adv. Res. Rep.* (2020) 44–52. doi:10.9734/ajarr/2020/v10i430253.
- [14] Automated control and autonomy, In: *Systems engineering for ethical autonomous systems*, Institution of Engineering and Technology, 2019, c. 41–81. doi:10.1049/sbra517e_ch3.
- [15] Y. V. Krak. Dynamics of manipulation robots: Numerical-analytical method of formation and investigation of computational complexity. *Journal of Automation and Information Sciences*, 1999, 31(1-3), 121-128. doi:10.1615/JAutomatInfScien.v31.i1-3.
- [16] L. Hewing, K. P. Wabersich, M. Menner, M. N. Zeilinger, Learning-Based model predictive control: toward safe learning in control, *Annu. Rev. Control, Robot., Auton. Syst.* 3.1 (2020) 269–296. doi:10.1146/annurev-control-090419-075625.
- [17] E. F. Camacho, C. Bordons, Introduction to model based predictive control, In: *Model predictive control*, Springer London, London, 1999, pp. 1–11. doi:10.1007/978-1-4471-3398-8_1.
- [18] I. Krak, O. Barmak, E. Manziuk, A. Kulas Data Classification Based on the Features Reduction and Piecewise Linear Separation. *Intelligent Computing and Optimization*. 2020. Vol. 1072. Pp. 282–289. DOI: https://doi.org/10.1007/978-3-030-33585-4_28.
- [19] D. M. Haftor, R. Costa Climent, J. E. Lundström, How machine learning activates data network effects in business models: Theory advancement through an industrial case of promoting ecological sustainability, *J. Bus. Res.* 131 (2021) 196–205. doi:10.1016/j.jbusres.2021.04.015.
- [20] L. S. Pontryagin, *Mathematical theory of optimal processes*, Routledge, 2018. doi:10.1201/9780203749319.
- [21] R. Bellman, On the theory of dynamic programming, *Proc. National Acad. Sci.* 38.8 (1952): 716–719. doi:10.1073/pnas.38.8.716.
- [22] R. Picciotto, Evaluation and the big data challenge, *Am. J. Evaluation* 41.2 (2019): 166–181. doi:10.1177/1098214019850334.
- [23] Welcome to CVXPY 1.4 – CVXPY 1.4 documentation. URL: <https://www.cvxpy.org/>.
- [24] A framework for neural network control. URL: <https://github.com/asikist/nnc>.