Univerza
v Ljubljani
Fakulteta
za gradbeništvo
in geodezijo

Doktorand
AHMAD FARHAD SKANDARY

UPORABA DIFERENCIALNE EVOLUCIJE ZA KALIBRACIJO INTEGRIRANIH MODELOV RABE TAL IN PROMETA

Doktorska disertacija

DIFFERENTIAL EVOLUTION APPROACH TO LUTI MODEL CALIBRATION

Doctoral dissertation

INTERDISCIPLINARNI DOKTORSKI ŠTUDIJSKI PROGRAM
GRAJENO OKOLJE

Ljubljana, November 2023

Univerza
v Ljubljani
Fakulteta
za gradbeništvo
in geodezijo

Mentor: Prof. Dr. Marijan Žura, University of Ljubljana, Ljubljana, Slovenia.

Komisija za spremljanje doktorskega študenta:

Assoc. Prof. Dr. Peter Lipar, University of Ljubljana, Slovenia,

Assoc. Prof. Dr. Ilka Čerpes, University of Ljubljana, Slovenia,

Asst. Dr. Marjan lep, University of Maribor, Slovenia,

Ljubljana, November 2023

ERRATA

Page	Line	Error	Correction

»This page is intentionally blank.«

ACKNOWLEDGMENTS

I want to begin by expressing my appreciation to my supervisor, Prof. Dr. Marijan Žura, for his advice and ongoing support in both personal and scientific matters throughout my doctoral studies. The gracious support and insightful guidance of my supervisors made this study possible. I would also like to thank all the FGG-PTI members who provided me with facilities and consent to this end. I also want to thank the dissertation committee members for their evaluation, helpful criticism, and direction in enhancing the quality of this thesis. I appreciate the efforts and advice of all my professors, who helped me learn more and conclude this work.

In particular, I would like to thank Prof. Naser Moen from Herat University, Ms. Melita Inkovec, the former Slovenian Civilian Representative in Afghanistan, and Ms. Zorica Bukinac from the Ministry of Foreign Affairs for their support in helping me secure this fantastic opportunity to study in Slovenia from the very beginning.

I want to express my sincere gratitude to Ms. Indira Džopa, Ms. Doris Sattler, and the rest of the staff at SRIPS-Slovenia for all their assistance and support while I was studying here.

All my friends have my sincere gratitude. It is a blessing to have such incredible friends who have assisted, encouraged and supported me through all the struggles in those years. We could not have made it to our destination without their unwavering support.

Ultimately, but most significantly, I want to thank my family for their unwavering love and support. None of this could have been possible without your love and sacrifices, my beloved wife.

My parents' souls flew to the skies before I started my master's program, but their love and care throughout my life gave me patience in challenging situations and inspired me to keep going. For everything you provided me during my entire life, and especially now that I have such an outstanding accomplishment, I am truly grateful and appreciate you the most.

»This page is intentionally blank.«

BIBLIOGRAFSKO-DOKUMENTACIJSKA STRAN IN IZVLEČEK

UDK: **004.8:656.1:7.114(043)**

Avtor: **Ahmad Farhad Skandary, MSc**

Mentor: **Prof. dr. Marijan Žura, univ. dipl. inž. grad**

Naslov: **Uporaba diferencialne evolucije za kalibracijo integriranih modelov rabe tal in prometa**

Tip dokumenta: **Doktorska disertacija**

Obseg in oprema: **91 str., 20 pregl., 27 sl., 30 en.,**

Ključne besede: **LUTI modeli, Raba Zemljišča, Transport, Kalibracija, DE, PSO, GA, Hibridni Algoritem, MANE, RMSE**

Izvleček

Modeli LUTI (interakcija rabe zemljišč in transporta) so orodja za pomoč pri odločanju za simulacijo kompleksnih dinamičnih dvostranskih povratnih informacij med transportnimi modeli in modeli rabe zemljišč. Modeli LUTI ocenjujejo več scenarijev načrtovanja, da bi prišli do najustreznejših odločitev. Sprejemanje odločitev na podlagi modelov, ki niso kalibrirani, je lahko zavajajoče in celo napačno. Čeprav je kalibracija (ocena parametrov) ključna zahteva modelov LUTI, popolnoma avtomatizirani pristopi z uporabo večciljnih funkcij niso bili v celoti obravnavani in ni standardnega postopka za kalibracijo modela LUTI. Modelarji namesto tega uporabljajo običajne tehnike za kalibracijo določenega elementa modela ali oceno skupine parametrov modela z malo ali brez skrbi za globalno shemo.

Cilj doktorske disertacije je razvoj popolnoma avtomatiziranega pristopa globalne kalibracije z uporabo večciljnih funkcij. Za odpravo te omejitve je predlagan splošni pristop kalibracije za parametre modela rabe zemljišč z uporabo algoritma diferencialne evolucije (DE). Izvedena je bila globalna analiza občutljivosti za identifikacijo najpomembnejših parametrov modela rabe zemljišč. Ti parametri so bili nato kalibrirani z uporabo algoritma diferencialne evolucije s korensko povprečno kvadratno napako (RMSE) in povprečno absolutno normalizirano napako (MANE) kot večciljnimi funkcijami. Predlagana tehnika (algoritem DE) ponuja pet ključnih zmogljivosti za kalibracijo modelov LUTI, vključno z 1) globalno oceno namesto lokalne ocene, 2) upoštevanjem večciljnih funkcij, 3) nenehnim izboljševanjem rezultatov, 4) enostavno prilagodljivim, in 5) vključitev več parametrov v postopek kalibracije. Za testiranje učinkovitosti predlagane tehnike kalibracije je bil uporabljen model rabe zemljišč TRANUS. Validacijo in konsolidacijo pristopa smo testirali na podlagi konvergence, minimiziranja napak in razmerja med modeliranimi in opazovanimi podatki s primerjavo z dvema dobro znanima optimizacijskima tehnikama, genetskim algoritmom (GA) in optimizacijo roja delcev (PSO). Naši poskusi kažejo, da je z uporabo algoritma Deferential Evaluation predlagani pristop presegel tehnike GA in PSO ter zagotovil najbolj stabilne in raznolike rešitve.

BIBLIOGRAPHIC-DOCUMENTALIST INFORMATION AND ABSTRACT

UDC:	**004.8:656.1:7.114(043)**
Author:	**Ahmad Farhad Skandary, MSc**
Supervisor:	**Prof. dr. Marijan Žura, univ. dipl. inž. grad**
Title:	**Differential evolution approach to LUTI model calibration**
Document type:	**Doctoral dissertation**
Notes:	**91 p., 20 tab., 27 fig., 30 eq.,**
Keywords:	**LUTI models, Land Use, Transportation, Calibration, DE, PSO, GA, HYBRID Algorithm, MANE, RMSE,**

Abstract

LUTI (Land Use and Transportation Interaction) models are decision-making aid tools to simulate complex dynamic bilateral feedback between transportation and land-use models within a territory. LUTI models appraise several further planning scenarios to arrive at the most appropriate decisions.

Making decisions based on the models that are not calibrated or calibrated properly might be misleading and even incorrect. Although calibration (parameter estimation) is a crucial requirement of LUTI models, fully automated approaches using multi-objective functions have not been fully addressed. There is no standard procedure for LUTI model calibration. Modelers instead use conventional techniques to calibrate a specific element of a model or estimate a group of model parameters with little or no concern for a global scheme.

This thesis aims to develop a fully automated global calibration approach using multi-objective functions. In order to overcome this constraint, a novel calibration methodology is introduced for the parameters of the land-use model, using a Differential Evolution (DE) algorithm. A global sensitivity analysis was performed to identify the most critical land-use model parameters. These parameters were then calibrated using the differential evolution algorithm with the Root Mean Square Error (RMSE) and Mean Absolute Normalized Error (MANE) as standard statistical metrics to measure the goodness of the proposed calibration approach. The proposed technique (DE algorithm) offers five critical capabilities for calibrating LUTI models: 1) global estimation, prioritizing over local estimation, 2) accommodating multi-objective functions, 3) continuously enhancing results, 4) easy adaptability, and 5) incorporation of multiple parameters in the calibration process. The performance of the proposed calibration technique was assessed using the TRANUS land-use model. The approach was validated and consolidated, evaluating convergence, error minimization, and the ratio between modeled and observed data. These assessments involved comparisons with two established optimization techniques: Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). Our experiments indicate that employing the Differential Evaluation algorithm resulted in the proposed approach outperforming both GA and PSO techniques. The Differential Evaluation algorithm provided superior performance and demonstrated excellent stability and diversity in solutions.

TABLE OF CONTENT

KAZALO

LIST OF FIGURES

KAZALO SLIK

LIST OF TABLES

KAZALO PREGLEDNIC

ABBREVIATIONS AND SYMBOLS/ OKRAJŠAVE IN SIMBOLI

Atrac.Fac	attractor factor
h	Shadow Prices in Tranus Model
δ	elasticity parameter
b	relative weight of sector
β	dispersion parameter of multinomial logit model
λ	factor that regulates the relative importance of prices versus transport disutility
X	Production
P	Price
CMA	Covariance Matrices Adaption
CR	Crossover Rate
C_1	Personal Learning Coefficient
C_2	Population (Global) Learning Coefficient
D	Problem Dimension
DE	Defferebtuak Evolution Algorithm
DEGA	Hybride DE and GA algorithms
DEPSO	Hybride DE and PSO algorithms
EA	Evalutaion Algorithm
EC	Evaluation Computation
μF	Mutation Factor
G	Number of Genration
GA	Genetic Algorithm
GADE	Hybride GA and DE algorithms
GUI	Graphical User Inrerface
It	Iteration Number
ITS	Intellengent Transportaion System
LUTI	Land Use and Transprtaion Integration models
MAN	Mean Absolute Normalized
MANE	Mean Absolute Normalized Error
NP	Number of Population
PSO	Particle Swarm Optimization
PSODE	Hybride PSO and DE algorithms
RMSE	Root Mean Square Error
W	Inertia Weight

»This page is intentionally blank.«

1 INTRODUCTION

Land use and transportation interaction are fundamental concepts in the study of land development and the formulation of transport interconnections. In most scenarios, land use and transport planning are conducted separately, which means that the effect of any change in transportation policies on land use patterns is frequently ignored. Urban sprawl is one of the consequences of neglecting such bilateral impacts in the planning process [1] and may lead to further incorrect assessments in decision-making.

LUTI (Land Use and Transportation Interaction) models are designed to predict the interrelations between economic growth and transport demand and vice versa (for more information, see [1]). LUTI models have been used to examine the impact of transport and land-use policies, such as the implementation of transportation infrastructures (e.g., Highway development, underground systems), dwelling and business improvements, improvement of public transport and fare changes, the expenses of private transport and the development of socio-demographic and economic scenarios as well [2]. According to Hunt et al. [3], such models must be integrated comprehensively and operative. These three elements indicate that the model should (1) adequately reflect the connections between transportation and land use, and contrariwise, it should also (2) simulate dwelling and professional location preferences to compensate for a wide range of spatial phenomena, particularly the development of land use (comprehensiveness), and finally, (3) there should be at least a single application for policy analysis of a metropolitan region.

1.1 Motivation

The evolution of urban land use and transportation infrastructure management has historically shaped city development. Numerous studies in the previous century have centered on these interconnected aspects, particularly emphasizing their relationships. LUTI (Land-Use and Transportation Integrated) models portray the multifaceted connections between land use patterns and the supply and demand of transportation within a region. They are primarily used to assess several alternative planning scenarios by modeling their likely implications on land use and transport patterns. LUTI models attracted the attention of researchers since the 1950s; researchers interested in modeling the complicated economic relationships in the urban area have turned to LUTI modeling; a comprehensive review of the evolution and history of LUTIs by Wegener can be found here [1]. LUTI models are multidisciplinary and incorporate econometrics, demographics, and transportation engineering. The models serve as the foundation for several simulation applications. Simulation findings must match real-world observations to be used as a framework for theoretical investigations of real-world assessment and prediction. The goal of LUTI simulations is to forecast the evolution of an urban region over a several-year timeframe, using presumptions in scenarios or strategies explored in particular research. Finally, the purpose is to analyze these strategies or techniques and choose the optimal alternative based on the simulation findings. LUTI models, like any other simulation model, include several parameters that reflect the properties and performance of its elements and their interconnections. Because they depend on both presumptions and accuracy of the information, like any other numerical model, they are fundamentally unreliable, and no scientific principle is available that guarantees the LUTI model's reliability. As a result, there are no impartial models or objective standards that can be used to determine if one configuration is better than the other [4], [5][6].

Furthermore, while these models will not be able to "meet all basic scientific standards" and be utilized for theory testing, they will be able to produce "robust but contingent knowledge" [7]. In this regard, calibration and validation should be considered. Several model calibration stages are required to guarantee that the proposed approach is a satisfactory approximation of reality under particular criteria described in the analyzed scenario, particularly for long-term forecasts[8].

1.2 Research Objectives and Hypothesis

Despite LUTI models being more widely used, their adoption is being restricted due to a lack of trust in their outcomes. It is not easy to make trustworthy projections with such complexity and scale. As a result, calibration and validation procedures are at the forefront of contemporary LUTI research. It requires estimating numerous parameters to set up LUTI models that can accurately duplicate the target region's actual data. Due to the lack of a global estimating procedure, most current LUTI calibration techniques are semi-automated and rely on single-objective functions and local estimation techniques. An Automatic and Global calibration approach is a highly desirable goal. Although the studies reviewed in the pieces show significant and influential efforts toward this end, as the main objective, this research aims to develop an Automatic and Global Calibration Approach for the LUTI models using a Multi-Objective Optimization Technique. To do so, the Differential Evolution (DE) algorithm, which was developed by Storn and Price [9], one of the most potent Evolutionary Algorithms (EAs), has been selected as an optimization technique, and the Root Mean Square Error (RMSE) and Mean Absolute Normalized Error (MANE) have been employed as standard statistical metrics to measure the goodness of the proposed calibration approach. DE was used because of its simplicity in terms of programming, fast convergence, global estimation, and the ability to find optimum solutions in almost every iteration. The land-use and activity model of TRANUS [10], [11] (one of the well-known open-source LUTI models) has been selected to test the suggested calibration technique. TRANUS is a standard structure for modeling sustainable land use and transportation at both the urban and regional scales. Two sub-modules are combined in TRANUS. (1) a land use and activity module that simulates a spatial economic system by analyzing activity locations and economic sector relationships, and (2) a transportation module that estimates the utilization of the transportation network and related disutility. Improving the TRANUS land-use module by calibrating the involved parameters is a sub-goal of this thesis. Therefore, only the land use and activity module of the TRANUS are considered in the calibration process.

In this research, the following **hypothesis** will be tested:

- **Using the Differential Evolution (DE) algorithm would improve the calibration of LUTI models.**

1.3 Overview of the Dissertation

Beginning with a comprehensive literature assessment of LUTI models and current calibration procedures, this investigation set out to accomplish its goals. Section 2 provides an overview of LUTI models, their applications, current calibration techniques, optimization methods, and a detailed examination of the chosen algorithm (DE) and the target LUTI model, TRANUS.

Section 3 outlines the methodology employed for achieving an Automatic and Global calibration approach. The mathematical foundation of the TRANUS land use and activity model, pertinent to this study, is elucidated. Subsequently, a sensitivity analysis employing Sobol indices is conducted to identify the most influential parameters of the target model. Objective (cost) functions are defined using Mean Absolute Normalized Error (MANE) and Root Mean Square Error (RMSE) as error metrics.

Continuing in Section 3, the calibration and optimization techniques employed in this study, including DE, GA, PSO, and Hybrid (GADE and PSODE), are customized based on the scenario provided by the TRANUS model. The economic parameters, identified through sensitivity analysis and shadow prices, are then calibrated using the specified optimization methods.

Section 4 presents the calibration results of the case study and discusses and compares the techniques utilized. Section 5 shows the dissertation's conclusion with a summary of limitations and suggests potential avenues for future studies.

2 BACKGROUND

Land Use and Transportation Interaction models are computational frameworks designed to integrate both land use and transportation systems, enabling the analysis of their dynamic interrelationships. These models aim to simulate the impact of alterations in land use patterns on transportation demand and, conversely, how transportation infrastructure influences decisions regarding land use. Typically, LUTI models employ mathematical, statistical, and simulation techniques to investigate the complex interactions between land use and transportation systems. They draw upon diverse datasets, including census records, land use maps, transportation network data, and various spatial-temporal datasets.
Widely applied in urban and transportation planning, LUTI models serve as valuable instruments for policymakers and planners. They facilitate the evaluation of potential consequences stemming from diverse land use and transportation scenarios. For instance, these models enable an assessment of how new transportation infrastructure projects, such as highways or public transit systems, might impact cities' spatial organization and their inhabitants' travel behavior.

2.1 LUTI models

This section focuses on the calibration of LUTI models and provides an overview encompassing their historical evolution, diverse applications, and contemporary calibration techniques. LUTI models stand as an integration platform, combining theory, data, and algorithms to capture the intricate interaction between the foundational components of urban areas: the transportation and land use subsystems. [12].
Land-use models are used to forecast demographic and economic indicators for land-based activities. These measurements characterize a particular metropolitan area's population (typically in terms of income and employment) and built-space environment (e.g., floor space). Predicting traffic patterns on a transportation network is done with the use of travel models. This class of models seeks to replicate travel patterns as a function of human activities (typically in terms of land uses) as well as transportation network features (commonly considered in terms of accessibility). Integrated land use and transportation models simulate the interactions of land use and transportation systems. In general, feedback methods are used to replicate this interaction.

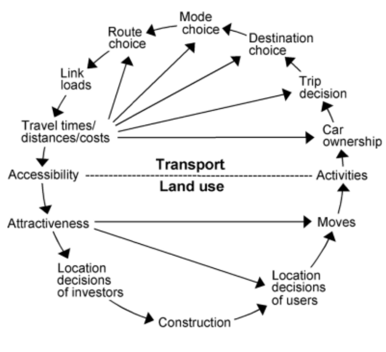

Figure 2-1: Feedback cycle between travel activities and land use [1]
Slika 2-1: Ciklus povratnih informacij med potovalnimi dejavnostmi in rabo zemlje

As illustrated in Figure 2-1, The feedback loop in LUTI models allows for considering the impacts of transportation policies on land use patterns and vice versa. For example, a new transportation infrastructure project, such as a new metro line, can influence the location and density of land uses along the route. In contrast, changes in land use policies, such as the introduction of zoning regulations, can affect travel patterns by changing the accessibility and connectivity of different city areas.
Many models are already accessible for simulation and planning exercises, and several writers have proposed categories that bring together distinct models based on various characteristics. Wegener[1] employed nine critical criteria to classify more than 20 models published in the literature: comprehensibility, structure, theoretical foundation, methodologies used, dynamics, data necessary, calibration, operationality, and application. Waddell and Ulfarsson [13] developed a LUTI classification based on the theoretical approaches established in LUTI modeling over the previous 50 years. Figure 2-2 represents a classification of LUTI models discovered in the literature according to the historical evolution of the main theoretical modeling paradigms. It is based on their core hypothetical nucleus and the temporal generation to which they belong.

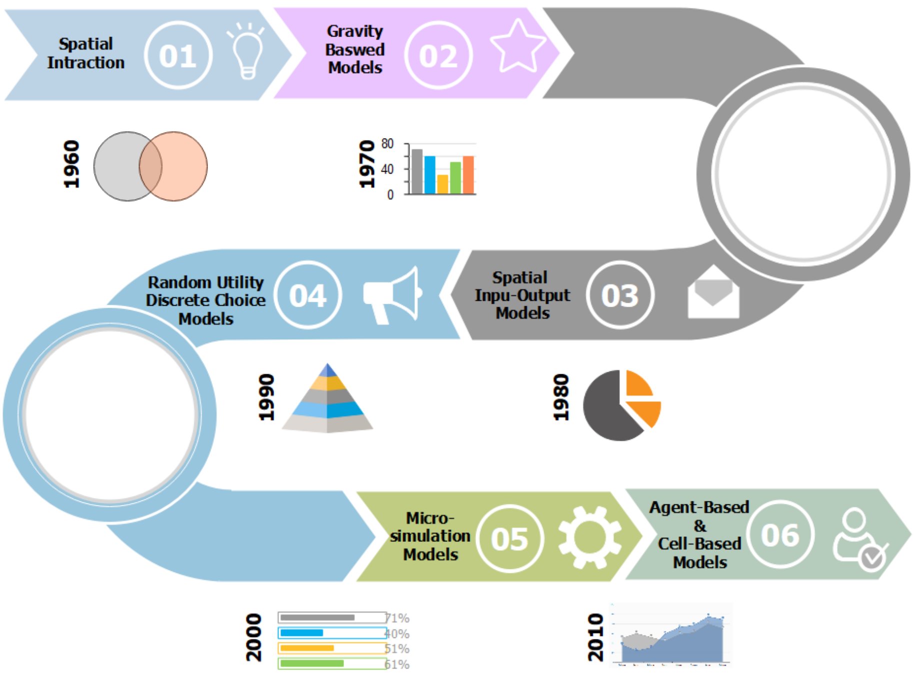

Figure 2-2: Chronological development of LUTI models [14]
Slika 2-2: Kronološki razvoj modelov LUTI

Based on this historical development and generation discovered, the following classifications are presented by Coppola et al. [15]:

1. Models introduced in the 1960s and 1970s are known as first-generation models. According to their theoretical foundation for running simulations, they may be split into three categories.

a. Lowry [16] established an interaction model for modeling population and economic activity areas, backed by economy foundation theory [17]. The theory of spatial interaction or Wilson's extension of statistical mechanics is the foundation for spatial and gravity models [18].
b. Optimization-based mathematical programming models. This approach minimizes or maximizes a specific objective to simulate agent behavior. Herbert and Stevens (1960) created the standard model of this kind, which approximated the behavior of the residential location market based on Alonso's theory of aggregated rent maximization. TOPAZ (Technique for Optimal Placement of Activities into Zones) is another example of this model, which calculated activity placements based on transportation cost reduction and urban growth [19].
c. INPUT/OUTPUT matrices-based models This model replicates the metropolitan or regional economy by utilizing the input/output matrix approach created from the work of [20]. MEPLAN is an excellent example of this sort of paradigm. [21], [22].

2. In the 1980s and 1990s, second-generation models debuted. These models are based on McFadden's work on random utility theory [23]. This generic type may be subdivided further into a simulation of land markets based on [24]. Another example of the second-generation modes developed by Martinez[25] is the Santiago Land use mode “MUSSA.”
3. Models from the third generation first arrived in the second part of the 1990s. These are highly disaggregated models, often called microsimulation models in some circumstances [14]. They are dynamic, meaning the answer to their simulations does not attain total market equilibrium. URBANSIM, created by Waddell and associates at the University of Washington, is one of the more well-known and commonly used models [13][26].

It is vital to remember that the three generations of models are still being researched, and none has been able to replace the others completely. Random utility theory is the most often utilized paradigm in site choice modeling for various urban actors. Even though numerous studies have proven that both techniques may yield identical findings under specific assumptions, this theory has essentially supplanted spatial interaction theory-based location models, which provided a more limited behavioral foundation [24].

LUTI models have been challenged in the realm of transport subsystem simulation because they employ approaches that are somewhat out of date. Many LUTI models still use the traditional four-stage sequential method, prompting some authors to advocate for adopting more current models that can be endogenous or exogenous to the rest of the LUTI simulator [27].

The primary purpose of Land Use and Transport Interaction (LUTI) models is to describe the complex interplay between land use and transportation in urban contexts. Models exclusive to a sector, such as those for transportation systems or urban development, cannot consider this relationship and, as a result, leave out an essential aspect of the tale. The goal of LUTI models is to close this gap and, in the end, provide better decision-making tools for long-term planning in cities and regions. The first LUTI models came in the 1960s. The LUTI models' complexity, along with the computing constraints of the time, put a halt to their progress. However, interest in LUTI models revived in the 1960s, and their quantity and complexity have gradually increased. This area also makes extensive use of micro-simulation. While spatial economics models may be more technically challenging to manage owing to the necessity of finding an equilibrium for a complicated collection of parameters and equations, activity-based models often have more parameters and more significant data requirements for instantiation. Aside from these distinctions, all LUTI models have various conditions, including, as with most models in general, the necessity for calibration (parameter estimation) techniques to instantiate them and validation ways to explain their operational capacity.

2.1.1 TRANUS

Tomás de la Barra [10] designed TRANUS, an op-source, widely used LUTI model. The TRANUS model is a widely used LUTI model. It incorporates various modules representing transportation systems, land use patterns, and economic factors. The model allows for analyzing land use and transportation interactions and can support urban planning and policy decision-making.

Two modules are connected and act as an input-output model in TRANUS. The activity model, which represents the interactions of numerous economic sectors during a specific period, simulates the spatial financial system. These interactions result in a demand for transportation, which is then sent into the network via the transportation module. It is specifically developed to simulate the likely consequences of various projects and policies in cities and regions and evaluate the outcomes from socioeconomic, budgetary, and environmental perspectives. The most valuable feature of the TRANUS system is the method in which all components of the urban or regional system, such as activity location, land use, and transportation, are tightly integrated.

2.1.2 Theory and Structure

TRANUS provides a platform to evaluate and model the integration of land use and transportation models. It can be applied on a local, regional, or national level. The study area is separated into spatial zones and economic sectors, with the essential notions of the original input-output model modified and given a spatial dimension. It is a macroeconomic equilibrium type model that combines two modules: (1) Land use and activity module to simulate a spatial economic system by analyzing activity locations and economic sectors relationships, and (2) Transportation module to calculate the usage of the transportation network as well as the related disutility. The two modules of TRANUS use random utility theory, e.g., discrete choice logit models for the designation of activities and land use, such as activity-location, land-choice, multi-modal path choice, and assignment. Each module is then run again until all regions' production and consumption needs are satisfied and equilibrium is reached. [28]. Significant theoretical developments have been produced in activity location interaction and transportation in the last several decades. Several formal theories utilizing quantitative approaches have improved the comprehension of urban and regional systems.

Figure 2-3 depicts the essential components of both modules. Within the land use and activity module, the spatial economic system is simulated by modeling the locations of activities and the relationships between economic sectors during a specific period. Conversely, the transportation module distributes and allocates the travel demand generated by the activity model to the transport supply. Both modules are integrated as an input-output model. Economic and geographical connections between activities, transportation, and the real estate market result in people and freight movements. In turn, the accessibility provided by the transportation system impacts the location and interaction of activities, as well as land rent. The land-use module must establish an equilibrium between supply and demand and the balance between the paid amount and the generating cost of each economic sector. The transportation module receives the transport demand as input and equilibrates the transport system to meet that need. Both modules are run continuously since a general equilibrium state is reached.

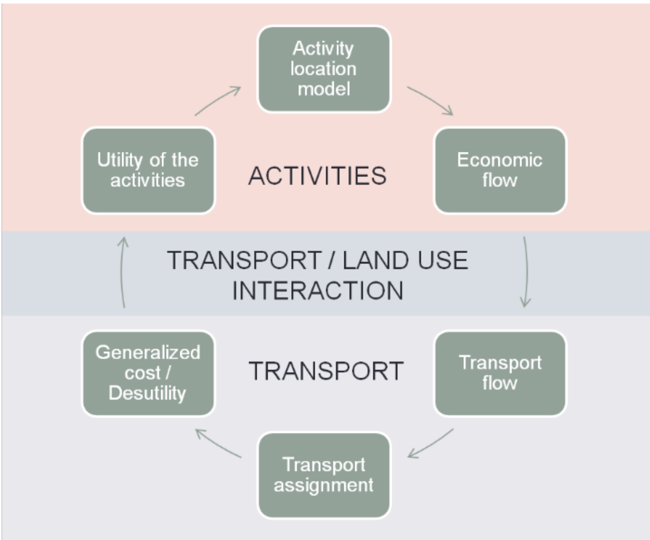

Figure 2-3: Fundamental Principle of TRANUS LUTI model [29]
Slika 2-3: Temeljni princip modela TRANUS LUTI

To calculate the costs and disabilities of transportation that will impact the activity model during the subsequent simulation times, the model first converts economic flows into transport flows before simulating a system of feedback loops to calculate the costs and disutility of transportation. According to the circle depicted in Figure 2-3, transportation and land use in Tranus will interact and have an integrated impact on one another. The initial input-output model in Tranus has been generalized to all sectors involved in urban dynamics, including land, activities, people, and transportation. As a result, the spatial component has been introduced and integrated with the transportation system [30].

2.1.3 Land Use and the Activity Module

The land use and activity module calculates the outputs and consumptions for a zone during a particular time and the flow demands generated by this activity. The transportation module receives these flow needs. The transportation network then allocates travel flows to the network based on travel demand when the associated trips are produced. In turn, the accessibility provided by the transportation system affects the location and interaction of activities via transportation disutility, which also impacts the land. The standard framework of the input-output model is used as a starting point, with final demand, intermediate demand, and main inputs. The destination of output, which often includes private consumption, government consumption, exports, and investment, is known as the final demand. The economic system must create the amounts necessary in each sector, necessitating intermediate inputs. These intermediate inputs, in turn, need other inputs, resulting in a lengthy cycle of production and consumption. Primary inputs and intermediate information are necessary and often include salaries, earnings taxes, and imports. Total output in the economic system equals the sum of all final and intermediate demand. Total production also equals the sum of all medium and main inputs [31].

It is necessary to distinguish between two different kinds of economic sectors: (1) transportable sectors and (2) non-transportable sectors. The critical distinction is that transportable sectors may be created and consumed in several locations, but non-transportable sectors can only be consumed where they are generated. For example, coal production in several places may fulfill the steel industry's requirement for coal. Land or structures that must be consumed wherever they are produced could be the most common non-transportable sectors. Non-transportable sectors, such as homes and net gross floor spaces, usually connect to real estate. Transportable sectors might include all types of employment and inhabitants. Explanations and details can be found in [11], [32].

As a result, transportable sectors produce flows, whether of commodities, services, or people. Transport infrastructure must be present for such movements to be viable, which adds transportation expenses to production costs. Non-transportable industries do not produce flows and do not utilize transportation. Typically, economic sectors are divided into three categories: land or floor space, households, and industries. The land is often divided into two or three types of residential floorspace (e.g., detached houses, flats, and mobile homes) and commercial floorspace (e.g., offices and markets). Households are often categorized by socioeconomic status, which is determined by income or the household structure. Industries (e.g., main products are for export), services (schools, colleges, institutions), and commerce are examples of business sectors.

Several parameters and functions represent the behavior of the various economic factors and their equilibrium achievement. Elasticity parameters ($\boldsymbol{\delta}$), sector weight, dispersion parameters ($\boldsymbol{\beta}$), initial attractor of each zone, attractor factors concerning sectors, and transport disabilities ($\boldsymbol{\lambda}$) are the main parameters involved in the TRANUS models. A global sensitivity analysis is performed for these parameters during the development of the proposed calibration approach in this thesis.

A set of characteristics and expressions that are found in TRANUS mathematical equations are presented here:

Exogenous Production: In the conventional input-output paradigm, the production not induced by internal consumption equals final demand. Because the location of this production is independent of other factors in the study region, it may be supplied to the model or allocated using spatial distribution functions.

Induced Production: Production created inside the study area for consumption by internal or external sectors. The industries that require it dictate the growth and placement of induced production.

Exogenous demand: The exogenous request is assigned to zones with induced order.

Induced Demand: Final or intermediate demand drives production.

Consumption Cost: It denotes the unit cost of a consuming sector. It is determined by the product price or expense in the production zone and the unit cost of transportation from the production zone to the consuming area.

Production Cost: A sector's unit cost of production in a manufacturing zone. It is determined by the total consumption costs of all its inputs plus the value-added.

Value Added: The value of a production unit is obtained by adding the value of capital and labor to the value of all other input commodities. Examples of value-added include payments to money (rent), delivery (salaries), government taxes or subsidies, costs on capital equipment, and so forth.

2.2 Calibrations of the LUTI models

Calibration of LUTI models is adjusting model parameters to ensure that the model outputs match observed data. It involves comparing model outputs to real-world data and iteratively adjusting the parameters until a satisfactory fit is achieved. Calibration is essential to improve the accuracy and reliability of the model's predictions and enhance its usefulness in decision-making processes. Its difficulties are based on time, resources, and data, such as theoretical, practical, and methodological issues [33]. As the main interest of this study is to develop a calibration approach for LUTI models, it is crucial to overview the existing calibration techniques of the current LUTI models.

Calibration (parameter estimation) is the most crucial factor in LUTI models [36]. This refers to estimating and adjusting model parameters using a numerical method to minimize differences between actual and modeled data. However, econometric ad-hoc procedures and trial-and-error techniques have been conventionally used to calibrate LUTI models [37]. Model calibration and validation are complicated processes that face theoretical, methodological, and practical challenges in terms of time, resources, and data; however, for end users to have confidence in the model, they must be comprehensible[33]. Because of the model's interactions, a modest change in one parameter might result in considerable changes in the model's outputs. In such circumstances, calibration is crucial since it aids in estimating ideal values for specific parameters, resulting in a more robust model [38].

LUTI models are gaining popularity, but their lack of accuracy keeps them marginal, especially at larger scales. Because of the lack of accuracy, simulation findings cannot be fully confirmed without calibration, and the calibration of the model should be handled with extreme caution to improve simulation accuracy [39]. One of the main weaknesses of large-scale models, according to Lee [40], is the lack of accurate and efficient ways of calibrating their parameters, identifying the values of the parameters in their equations that resulted in the most extraordinary fit between the model outputs and real-world observations. Because LUTI models are numerical approaches, they are inherently uncertain due to the theoretical assumptions they are founded on and the quality of the data they employ. There is no physical law to confirm the model's credit and validity, and no absolute and comprehensive technique to calibrate and validate either basic or complicated LUTI models has been established or created.

To calibrate LUTI models, many parameters must be estimated, which will be time-consuming depending on the estimation technique used. Calibration is a collection of approaches for defining parameters with realistic values, precise modification, and attempting to fit the model data as closely as possible to the observed data [41]. Calibration can refer to constructing a model structure, including selecting a theoretical model, the operational forms of functions and explanatory variables, and, ultimately, assessing model parameters [35].

Despite significant advances in econometrics, optimization, and computer algorithms, the problem persisted, as insisted by Wegener[1][27]: "There has been almost no progress in the methodology to calibrate dynamic or quasi-dynamic models. In the face of this dilemma, the insistence of some modelers on 'estimating' every model equation appears almost an obsession. It would probably be more effective to concentrate instead on model validation, i.e., comparing model results with observed data over a more extended period".

Most articles on LUTI models and their applications do not fully describe the calibration technique. They usually only offer instructions for calibration, even if this work takes months or years and tremendous resources, and very little guidance is provided on how to instantiate one of these models. This thesis focuses on developing a global and automatic calibration strategy for LUTI models, so the models in which such approaches have been applied or created are discussed in this section.

The optimization approach has been widely utilized as an econometric methodology to calibrate specific portions of the LUTI models; for example, maximum-likelihood optimization is a recurring strategy to calibrate the microsimulation sub-models that many LUTI models share. The trial-and-error process of looking for best-parameter values by exhaustively running the model over various parameter values or combinations constitutes a crude approach to model calibration. Techniques must be used to identify parameter values that best meet the fitness test criteria of an urban model to calibrate it. For example, it may be determined that the optimal parameter values may be identified by minimizing the sum of the squared variances between forecasts and observations [42]. Boittin et al. [39] used Particle Swarm

Optimization (PSO) to calibrate LUTI models to overcome the limitations of existing methods. They concluded that many PSO variants are emerging, and it is unclear which one offers better calibration results.

MEPLAN is a multifunctional software system that may be used for various land use, transportation, and economic planning projects. It is based on the expertise of Marcial Echenique & Partners (ME&P), a Cambridge-based research and consultancy firm, in applying mathematical models to actual planning challenges over time [43]. Abraham and Hunt [21] developed a semi-automatic calibration approach for MEPLAN based on the minor square optimization technique. They proposed a simultaneous and sequential calibration approach that was later used in the location choice model for nested logit parameters.

PECAS[44] is a method for modeling geographic economic systems that are broadly applicable. It is intended to simulate the land use component of interactive modeling systems for the land use and transportation model, which MEPLAN inspires. Depending on the technical coefficients, the model system is based on a quasi-dynamic equilibrium structure with flows of exchanges, including products, services, and labor, from production to consumption. Nested logit models that include exchange pricing and transportation disutility are used to analyze trade flows from production to zones of exchange and from exchange zones to consumption. The calibration of the PECAS model was also based on the minimization of the least square technique. The Pirandello is a French LUTI model designed primarily for the Vinci firm by Jean Delons [45]; it offers a theoretically based yet approachable framework for discussing and analyzing land-use and transportation strategies. Pirandello seeks to enhance the housing market's representation while delivering information that is simple to understand. Gradient descent is routinely used to calibrate parameters or groups of parameters, enabling partial alterations to the model [46].

Paul Waddell [26] created the popular agent-based model UrbanSim, incorporating socioeconomic change modeling and household composition. Later P. Waddell et al.[13] applied UrbanSim on a detailed land-use simulation model system and its integration with a regional travel demand model in Utah's Greater Wasatch Front area. With specific models, UrbanSim replicates the geographical distribution of households, employment, real estate development, and real estate values annually using a dynamic disequilibrium modeling technique. Multinomial logit simulates household and job location decisions, whereas ordinary least square (OLS) regression is used for real estate development and valuation[47]. The approach for estimating model sensitivity and calibrating the complete model is provided here[48].

The land use framework UrbanSim was used to model housing prices in the Lyon urban region. A nine-year back-casting period is used to calibrate the home price model. When used in simulation, the calibrated model produces pricing dynamics comparable to those seen in the heart of Lyon. The model's capacity to reflect changes in employment accessibility on price dynamics is demonstrated via sensitivity analysis [47].

Stephen Putman developed and applied the ITLUP [49] framework at the University of Pennsylvania in Philadelphia, USA. ITLUP comprises two modules, DRAM and EMPAL, with more than a dozen active applications and more than 40 calibrations completed both within the USA and abroad; ITLUP is the most popular spatial allocation framework in the USA.[50].. ITLUP models distribute jobs and families to zones even if they cannot handle new employment or households. The present program addresses this constraint by reallocating excess allocation to zones that can accommodate additional jobs and homes. Another shortcoming of ITLUP is that the DRAM and EMPAL models are used sequentially, ignoring concurrent interactions between occupations and residences. Furthermore, because ITLUP does not consider land prices and commodity flows when distributing jobs and households, it misses essential linkages and variables of considerable relevance to planners, politicians, and the public [51]. A sensitivity analysis via Monte Carlo simulation is also included in this work, with very little detail concerning the calibration. Nonetheless, the black box methodology was determined to be the calibration method.

The MUSSA [52] LUTI model was created to estimate the expected placement of inhabitants and firms in metropolitan areas using bid-rent and market equilibrium. Macroeconomic assumptions are used to forecast population and business growth over time. The location of inhabitants and firms are then predicted by MUSSA utilizing a static demand-supply equilibrium with location externalities at a particular time in the future. The model comprises a sequence of nonlinear fixed-point equations, and the solution is found using an iterative gradient descent approach. Econometric approaches are used to calibrate MUSSA, providing the parameters needed by functions that reflect demand and supply behavior. [25]. For MUSSA, no automatic or semi-automatic calibration approaches have been proposed.

The following input-output model may be used to explain the calibration process implemented in TRANUS mathematically.

$$[X, H]^T = f(X^{act}, P^{act}, \rho) \qquad 2.1$$

Where H stands for the vector of adjustment parameters (shadow prices), which enables the correction of the initially provided prices by appointing appropriate values of unit production prices P, X denotes the matrix of computed production, and ρ represents the vector of economic parameters.

TRANUS calibration process uses data and economic parameters as inputs and adjustment variables (shadow prices $\boldsymbol{h}$) as outputs; this process is shown in Figure 2-4.

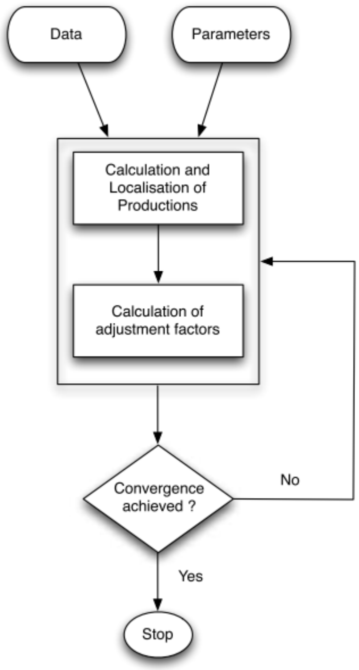

Figure 2-4: TRANUS calibration process[53]
Slika 2-4: Postopek kalibracije TRANUS

The land use and activity module equilibrates offer and demand repeatedly, as well as computing consumption costs and pricing. The transportation module, on the other hand, allocates the network's transport demand and computes the new transportation costs. This process is repeated until a general equilibrium state is discovered. The model has been used extensively throughout North, Central, South America and Europe.

NICOLAS PUPIER [29] developed the LUTI models of Belo Horizonte cities in Brazil using TRANUS as the LUTI model, although little detail is provided on how the calibration was done. TRANUS is also used for Lille in France. Ad-hoc processes and econometric methodologies were used as the calibration techniques by Fausto Lo Feudo [54]. Various econometric approaches and ad-hoc submodule calibration are used for several calibration methods. The purpose of optimizing using fundamental solvers is to get better parameter estimates. In all other cases, a highly skilled consultant does the calibration. (e.g., http://modelistica.com).

P. Dutta et al. [55] developed an algorithm to calibrate the LUTI models using maximum likelihood estimation. Furthermore, they examined the propagation of uncertainty during the calibration process of TRANUS using the Monte Carlo method. Then, a probabilistic verification methodology of the calibration process using a statistical hypothesis test was proposed. They noted that the error in the observed values of the outputs from the land-use module follows a Gaussian error. An assessment of TRANUS shadow prices and other parameters has been conducted by Capelle et al. [56], who also developed an optimization methodology for the partial calibration of TRANUS [6]. Here, shadow prices are price-correcting additive variables that compensate utilities to mimic base-year production. Estimating shadow prices is a part of the calibration of land-use models.

Nevertheless, it was discovered that the original TRANUS model with one shadow price per observation bore a risk of overfitting. In addition, Capelle et al. noted that determining and removing the unnecessary number of shadow prices may not be performed automatically and requires an expert eye. Later, Feudo et al. [13] proposed a semi-automatic process using non-linear optimization and curve fitting to calibrate the floor space substitution parameters, and they extended the technique proposed by Capelle et al. [61].

2.3 Optimization techniques

Optimization is a significant topic of mathematics with various subfields focused on problems with specific features that may be used to find efficient solutions. A comprehensive overview of current changes and prospective tendencies of optimization techniques have been provided here[62], [63].

Optimization is a branch of applied mathematics concerned with determining the extremal value of a function in a defined domain while considering various variable values. The following system of equations is the general form of an optimization problem, which is a strategy for optimizing the objective or cost functions:

$$\begin{aligned} min_x \quad & \boldsymbol{f}(x) & \\ & \boldsymbol{g}_i(x) \le 0 & i = 1,2,\dots,\mathrm{m} \\ & \boldsymbol{h}_j(x) = 0 & j = 1,2,\dots,\mathrm{n} \\ & \boldsymbol{x}_{kL} \le \boldsymbol{x}_k \le \boldsymbol{x}_{kU} & k = 1,2,\dots,\mathrm{p} \end{aligned} \tag{2.2}$$

In equation 2.2, the objective function is indicated by $\boldsymbol{f}(x)$, the constraints are denoted by $\boldsymbol{g}_i(x)$, $\boldsymbol{h}_j(x)$, and the decision variable vectors $(\boldsymbol{x})$ That fulfills the desired constraints, referred to as the viable solutions of the optimization model. Optimization techniques or algorithms answer the problems described in equation 2.2. The method determines the design variable values that produce the optimum objective function value while fulfilling all equality, inequality, and side constraints. The available optimization approaches can be classified in several ways.

2.3.1 Local Optimization

We are not interested in the local optimization technique in this work, so only a brief overview is covered here. The goal of function optimization is to find one of its extrema. Maximizing may be converted to minimization and vice versa by changing the overall sign of the supplied function. As a result, the terms minimization, maximization, and optimization are frequently used interchangeably. Finding a minimum value for convex functions is equivalent to finding the lowest feasible function value, and this value may be determined with a single local minimization from any starting point. However, a nonconvex function with several minima of distinct function values is encountered in many fascinating applications. Finding only one minimum, on the other hand, gives us nothing about the global optimum function value. Also, they discovered that the starting point of the optimization approach will highly influence local minimum. Gradient-based algorithms are used in the majority of local optimization strategies. Gradient-based algorithms are commonly utilized in engineering to solve various optimization issues. These methods are popular because they are efficient (regarding the number of function evaluations necessary to identify the optimum), can handle problems with many design variables, and require slight problem-specific parameter adjustment. These algorithms, however, have significant shortcomings, including the ability only to find a local optimum, difficulties addressing discrete optimization problems, and sophisticated algorithms that are difficult to implement quickly and are subject to numerical noise [64].

2.3.2 Global Optimization

Throughout history, generic optimization topics have played an essential role in engineering applications. Lagrange published the first significant work in optimization in 1797 [65]. Unlike local optimization, regardless of the beginning position, a genuinely global minimization should be able to determine the global minimum function value. In a non-convex situation, such deterministic global optimization offers beneficial information. Sometimes, problems have several optimum values, as illustrated in Figure 2-5 below by a simple one-variable function.

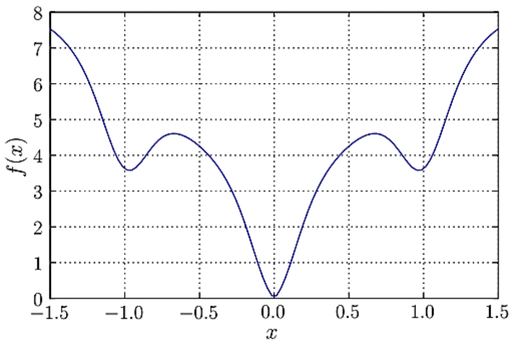

Figure 2-5: One-dimensional multi-modal function
Slika 2-5: Enodimenzionalna multimodalna funkcija

As illustrated in Figure 2-5, the minima at $\boldsymbol{x} \cong \mp 1$ are local (or relative) minima, whereas the minima at $\boldsymbol{x} = 0$, are global (or absolute) minima. Depending on which of these three points is reached first, the local algorithms outlined thus far will converge on either of these three places. When employing local optimization techniques, a multi-start approach is a straightforward way to deal with several local minima in the design space [66][64].

Global optimization algorithms have a significantly greater probability of discovering a model's global or near-global optimum value than the local algorithms covered thus far. It should be noted that no algorithm can guarantee general convergence on a global optimum. Therefore, referring to these algorithms as having global features may be more correct. International optimization techniques can be categorized as stochastic search (e.g., evolutionary computing) or deterministic algorithms. Several studies have examined the methodology and applications of deterministic global optimization, demonstrating the advanced state of research and the widespread usage of such approaches in practice[67], [68].

As we are interested in developing an automated and global optimization strategy for calibrating LUTI models utilizing the differential evolution (DE) algorithm, a member of the evolutionary algorithms (EAs) family, this thesis does not explore deterministic techniques. For decades, stochastic–heuristic global optimization methods have been used to solve much more significant problems at the cost of losing any assurance about the solutions or even the convergence behavior. Fundamental introductions to EAs and global optimizations may be found here[69], [70]. This thesis provides a brief sketch in the following sections for completeness.

2.4 Multi-objective optimization

Optimizing the objective function(s), which one or multiple equations can represent, is the final goal of an optimization problem. Most optimization methods involve a single objective function that seeks to maximize a specific benefit or income or minimize a particular cost or time. The optimal trade-off between two or more related objectives is optimized using multi-objective functions. Multi-objective optimization problems are frequent in engineering applications due to the multi-criteria decision complexity of most real-world situations. As the name implies, multi-objective optimization situations contain numerous objectives that must be maximized or minimized simultaneously and are frequently at odds with one another. Because evolutionary algorithms (EAs) deal with a set of candidate solutions, it appears reasonable to use them to identify a set of optimum solutions in multi-objective optimization problems. EAs have proven efficient in addressing multi-objective optimization issues [71].

2.4.1 Evolutionary Computation (EC)

Evolutionary computation is a method for finding optimal solutions to problems by iteratively testing several candidates or prospective solutions, selecting the "better" ones, modifying them, and producing new candidates to verify based on fitness values. Evolutionary computing (EC) is a recent search strategy that employs computational models of evolutionary and selection processes.

Evolutionary computation derives much of its terminology from genetics, cellular biology, and evolutionary theory since it is motivated by natural selection and genetics. A potential solution is an *individual*, while the *population* refers to the total number of individuals currently in the evolutionary computation system. Depending on the circumstances of the answer to a problem, this population may be subdivided into different population subgroups. The *genome* or *chromosome* is an individual's physical representation (encoding). Each chromosome comprises a series of *genes* or traits that characterize an individual. When individual solutions are adjusted to generate new candidate solutions, this is called *breeding*, and the new candidate solution is referred to as an *offspring* or a *child*. A potential solution is assigned a *fitness* value during *evaluation*, demonstrating the solution's validity in the context of a specific problem. A new *generation* is defined as the replacement of the present population by offspring. Finally, *evolution* reflects seeking the best choice [75].

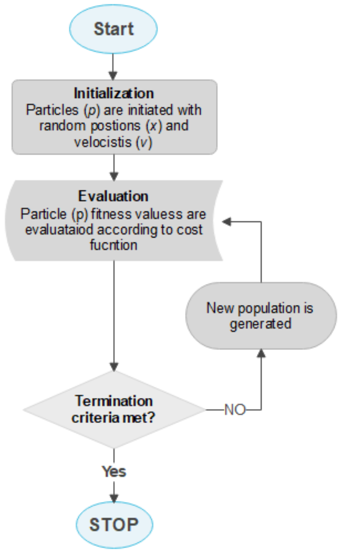

Figure 2-6: Evaluation Computations (ECs) General Flow-Chart
Slika 2-6: Splošni diagram poteka ocenjevalnih izračunov (EC)

As illustrated in Figure 2-6, an initial population of individuals (potential solutions) is formed before the valid evolutionary process begins. The initial population has traditionally been generated randomly, although numerous different initialization procedures have also been utilized (e.g., starting from a set of previously known or arbitrarily assumed solutions). Each member of the initial population is then examined and awarded a fitness value. The selection process picks a portion of the current population as parents to develop new individuals based on their fitness ratings.

When the selection system favors individuals with higher fitness values, the children produced are more likely to be competent. After selecting a set of parents, the new individuals are formed by duplicating them and using mutation technicians. The general form of evolutionary computation is shared among all its family members. It begins with the generation of individual populations at random and continues with evaluating the fitness of the individuals within the population. The best individuals are chosen to breed, resulting in the creation of a new population. The old population is then replaced with the new population, and the procedure is repeated during the fitness evaluation phase. The process repeats until a perfect candidate is found or all resources are depleted. The program returns the best-fitting individual it found during its runs.

Algorithm 2-1: General Evaluation Computation [75]

Initialization
    population $\boldsymbol{P} \leftarrow$ Initialize ($\boldsymbol{NP}$)
    Evaluation (individual $\boldsymbol{x_{best}} \leftarrow$ nil)
Main loop ($\boldsymbol{G}$)
    Repeat until the stopping criteria are met.
    For each individual $\boldsymbol{x_i} \in \boldsymbol{P}$

Evaluate fitness ($\boldsymbol{x_i}$)
If the fitness of $\boldsymbol{x_i}$ is assessed as optimal
Break and return $\boldsymbol{best}$
If $\boldsymbol{best}$ is nil or the fitness of $\boldsymbol{x_i}$ is better than the fitness of $\boldsymbol{best}$
$\boldsymbol{best} \leftarrow \boldsymbol{x_i}$
Population $\mathbb{Q} \leftarrow \phi$
Until $\|\mathbb{Q}\| = \|\phi\|$
$\mathbb{Q} \leftarrow \phi \cup \boldsymbol{Bread}\ (\boldsymbol{P}, \|\mathbb{Q}\| - \|\phi\|)$
$\mathbb{Q} \leftarrow \phi$
Return $\boldsymbol{best}$

The population size is determined by ($\boldsymbol{NP}$), and the value (***G)*** represents the number of *generations* or how many times the population is appraised and bred until the algorithm gives up. Population Initializing returns a set of ($\boldsymbol{NP}$) Initial individual, which is typically produced at random.
Fitness evaluation ($\boldsymbol{x_i}$) assigns a fitness grade to an individual based on the quality of prospective solutions. Breed ($\boldsymbol{P}, \|\mathbb{Q}\| - \|\phi\|$) selects individuals from the population $\boldsymbol{P}$ based on their fitness, duplicates them, and then uses a modification procedure to develop new candidate solutions. The breed produces an optimum of $\|\mathbb{Q}\| - \|\phi\|$ New individuals (typically just one or two) are introduced to the next-generation population $\mathbb{Q}$ [75].

2.4.2 Evolutionary Algorithms (EAs)

EAs are a stochastic search and optimization approach based on natural biological evolution principles [76]. EAs work with possible solutions that are updated based on two fundamental concepts. EAs have basic processes, yet they have shown to be resilient and have intense search and optimization approaches. However, these optimization issues typically need a considerable number of computational capabilities and include a significant quantity of unknown information. Evolutionary methods for handling multi-objective optimization problems have become famous as a study area in recent years. Among the various approaches proposed, three that are exceptionally comparable and popular are the genetic algorithm (GA) [77], particle swarm optimization (PSO) [78], and differential evolution (DE) algorithm [79]. While GA is more well-established due to its earlier introduction, the more recent PSO and DE algorithms have begun to receive increasing attention as multi-objective optimization techniques. This study's primary focus is applying and developing the DE algorithm as a multi-objective optimization technique for calibrating LUTI models. Because of their similarities, the Genetic algorithm and PSO optimization are used as comparative versions to validate the proposed calibration approach. Evolutionary algorithms are based on Darwin's theory of evolution [80], which outlines the survival of the fittest through natural selection and the enhancement of individual species' fitness. The natural selection of randomized individuals contributes to the search for the optimal chromosomal value in the universe of potential solutions. The algorithms designed under the umbrella concept of evolutionary computation are primarily focused on selecting a population as the initial possible solution. The quality of the original population continuously improves because of incremental processing of the current population employing evolutionary operators, including **crossover**, **recombination**, **selection**, and **mutation**. Various computing algorithms, such as **genetic algorithms**, **genetic programming**, **evolutionary strategies**, and **evolutionary programming**, have been developed based on this evolutionary principle and can address complicated problems where standard mathematical approaches are challenging to apply [81].

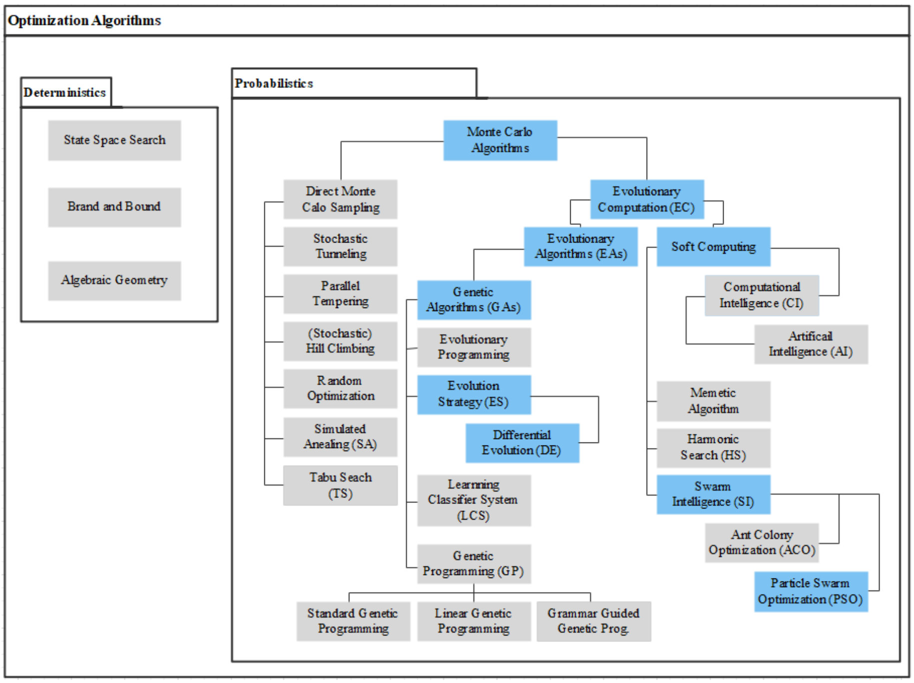

Figure 2-7: Classification of Optimization Algorithms [82]
Slika 2-7: Razvrstitev optimizacijskih algoritmov

2.4.3 Genetic Algorithm (GA)

Genetic algorithms (GAs) are evolutionary computing algorithms that employ methodologies influenced by natural evolution. Professor John Holland [77] of the University of Michigan devised and developed the first GA in the late 1950s and early 1960s, which tackles various optimization issues using biological genetic and evolutionary concepts. The GA algorithms have demonstrated their application power in resolving real-world matters, which are vague, complicated, and involve multimodal objective functions. The accompanying algorithms, such as simulated annealing and other guided random methods, are comparable to this optimization method. GAs use random search techniques to locate the global optimum of the solution. These algorithms outperform "gradient descent" approaches, which are susceptible to being stuck in local minima.

On the other hand, GA is distinct from pure random search algorithms in that they instantly look for the relatively "prospective" portions of the search space. GAs are suitable for discrete and noisy spaces and, as a solution, may be considered optimal. Complex circumstances, such as nonlinearity and shifting parameters, impose increased demands on the use of GA in land use research, with infinite issues [86] Holland [77] proposed that binary strings be used to represent the chromosomes. Crossover and mutation are genetic operators that create children from two sets of chromosomes. Crossover refers to the exchange of parts of genes between two chromosomal parents. The division of the chromosomes into two portions is determined at random. Mutation occurs when individuals in the genome are switched. The third operator is a random selection, in which the probability of being chosen is proportional to the individual's fitness (cost or objective function value). As a result, even the weakest candidate can be selected.

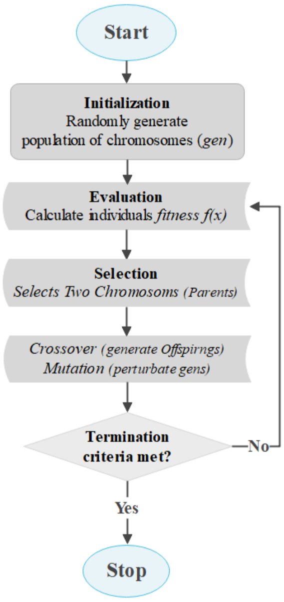

Figure 2-8: Genetic Algorithms (GA) General Flow-Chart
Slika 2-8: Splošni diagram poteka genetskih algoritmov (GA)

Figure 2-8 Illustrated the GA implementation process. Even if this is not true, the GA is built on one primary precept: "Good parents have better children.". Generating a random population of chromosomes (potential solutions) is the initial stage of the algorithm. The parents are represented by two chromosomes chosen from the population. The degree of *fitness*, *f(x),* of each chromosome in the population is used to evaluate its performance. The next stage is to choose two chromosomes from a new population to act as parents in a crossover process. Crossing the parent's chromosomes may result in children (*offspring*) based on the user's specified crossover probability. If there is no crossover, the offspring are identical to their parents. The mutation is a random process in which the genes inside the chromosomes are perturbed according to the mutation probability set by the user. Following the mutation process, new chromosomes are born and assessed by their *f(x)*. If the children's *f(x)* is higher than the parents', this pair of chromosomes is assigned to a new population during the performance evaluation procedure. If the parents outperform the children, the parents' chromosomes are introduced into the following generation. For the subsequent iteration, the best *f(x)* of children or parents will be included in the new population. Finally, the entire procedure is repeated until the termination criteria are met.

2.4.4 Particle Swarm Optimization

Particle swarm optimization (PSO) is a stochastic search technique inspired by the flocking/swarming behavior seen in nature. PSO's origins may be traced back to the use of particle systems in the graphical modeling of ambiguous objects. Eberhart and Kennedy [78] developed PSO based on these particle flocking systems. The swarm terminology is selected instead of flock since the behavior of the systems was consistent with swarm intelligence principles [87]. The PSO, established based on the regulations and laws of socially structured populations in nature, is one of the most exciting study fields within computational swarm intelligence. The swarm is made up of particles, which are individual agents. The population of particles is used in PSO optimization, representing the potential solutions overall in the search space. Particles move around the search area according to predetermined dynamics and eventually converge toward the best solution. Swarm-based algorithms have promised performance since they are efficient, resilient, and easy to implement [88]. The strength of PSO, in comparison to other Artificial Intelligence optimization approaches, lies in its ease of implementation.
Regarding success rate, solution quality, and convergence speed, the performance of several optimization techniques now utilized in the industry and their computing efficiency clearly shows that PSO outperformed other algorithms [89]. Each particle in PSO contains a memory component, distinguishing it from different naturally inspired evolution algorithms like genetic algorithms (GAs) or differential evolution (DE). In contrast to GA or DE, where population individuals cannot return to a previously preferred solution, memory is a crucial aspect of PSO. It permits particles to return to previously best-known solutions [90].

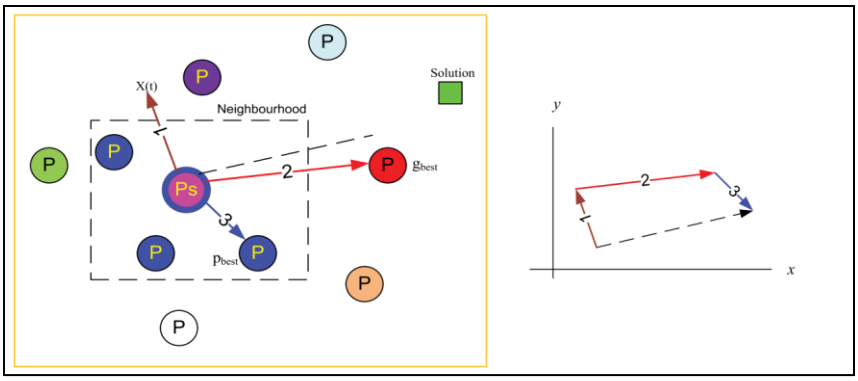

Figure 2-9: (a) Particle local and global best (b) Particle vector components [91]
Slika 2-9: (a) Lokalni in globalni najboljši delci (b) Komponente vektorja delcev

Figure 2-9 (a) describes the PSO approach, which involves a collection of possible solutions known as a "swarm of particles" generated in the search space with initial random positions and velocities. At the same time, the PSO vector components are seen during the updating process in Figure 2-9 (b). The current location's essential, local, and global best vectors are added to provide the updated position [92]. The revised position results from the summation of the primary vectors [83]–[85] of the current situation, local best vector, and global best vector. Every particle has its location and velocity at any specific time

[93]. Each particle in the search area attempts to find its solution to the problem to reach the "global best" solution. All particles have fitness values that are assessed by the cost or fitness function to be optimized and have updated values, as well as velocities that regulate particle movement.

PSO has outstanding application results; it has lately been utilized successfully in combination with various deformable models. Asl and Seyedin [94] use PSO instead of GA to implement the approach presented in [95], yielding the same precision in less time. Cruz-Aceves et al. [96] have introduced a new picture segmentation approach based on several active contours driven by particle swarm optimization. A multi-population PSO operates an operational contour model [97], stressing the model's capacity to adapt to geometries with substantial concavity.

Because of its performance in addressing unconstrained optimization problems, several researchers have employed PSO to solve restricted optimization problems in recent years. Liang and Suganthean [98] presented a novel dynamic multi-swarm PSO with a novel constraint to address constrained optimization problems. Krohling and Coelho [99]suggested a co-evolutionary PSO based on Gaussian distribution for handling restricted optimization by generating the acceleration coefficients using a Gaussian probability distribution. Pulido and Coello [28] developed a straightforward approach for dealing with PSO constraints. According to their stated method, if the particles evaluated are infeasible, the best particle has the lowest value in its normalized violation of conditions.

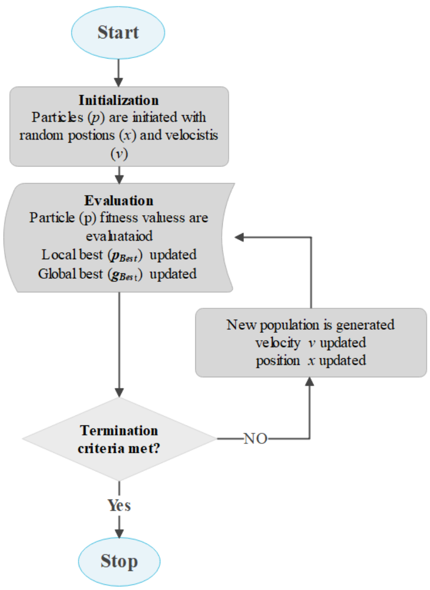

Figure 2-10: Particle Swarm Optimization (PSO) General Flow-Chart
Slika 2-10: Splošni diagram poteka Optimizacije Roja Delcev (ORD)

Figure 2-10 describes the general approach of the PSO, which initializes with a set of random particles that represent potential solutions, and then it updates generations to look for optima. Because the technique is iterative, the positions will vary with each time step. Furthermore, each particle will keep track of its optimum position. The particles are updated by the two best values at the end of each iteration. Local best (P_{Best}) is the best solution or fitness it has obtained so far, though, at which optimum values are stored throughout the process up to the current iteration. Global best (g_{Best}) is the other best value, the fittest position amongst all the other particles within the swarm [100]. Position and velocity change are estimated with equation 2.3, whereas equation 2.4 yields the new position of the target particle.

$$\boldsymbol{v}_i^k = \boldsymbol{w}\boldsymbol{v}_i^{k-1} + \boldsymbol{c}_1\boldsymbol{r}_{i1}^{k-1}\left(\boldsymbol{p}_i^{k-1} - \boldsymbol{x}_i^{k-1}\right) + \boldsymbol{c}_2\boldsymbol{r}_{i2}^{k-1}\left(\boldsymbol{p}_g^{k-1} - \boldsymbol{x}_i^{k-1}\right) \qquad 2.3$$

$$\boldsymbol{x}_i^k = \boldsymbol{x}_i^{k-1} + \boldsymbol{v}_i^k \qquad 2.4$$

Where:

- k represents the current iteration, while the prior iteration's index is $k-1$
- $i = 1, 2..., N$ is the population size
- c_1 and c_2 are the acceleration coefficients, commonly between [1, 2]
- r_{i1} and r_{i2} are evenly distributed random numbers between [0, 1]
- w stands for the inertial weight factor, which is normally between [0, 1]
- x_i^k represents the position of the particle i at the iteration k
- v_i^k represents the velocity of the particle i at the iteration k
- p_i^{k-1} represents the personal best position of particle i for the previous iteration.
- p_g^{k-1} represents the neighborhood's best position for the previous iteration.

2.4.5 Differential Evolution Algorithm

Storn and Price[79] presented a stochastic population-based evolutionary algorithm called Differential Evolution (DE). DE is a straightforward yet effective method for solving issues in continuous spaces. The DE method is a well-suited optimization technique due to its limited number of control parameters. Like GA, the DE algorithm has similar operators: crossover, mutation, and selection. The main difference between the GA and DE is the mutation scheme that makes DE a self-adaptive selection process. The primary advantages of DE over a typical GA include its accessibility, its efficient memory use, reduced computational complexity (it scales better when dealing with significant issues), and its lesser reliance on computing efforts (faster convergence) [104]. DE has several advantages over other evolutionary algorithms in that it is simple, easy to use, fast, and has a higher chance of discovering the global optimal solution for function optimization [79], [105]–[108].

Moreover, DE optimization is simple, fast, easy to use, very easily adaptable for integer and discrete optimization, quite effective in nonlinear constraint optimization, including penalty functions, and valuable for optimizing multi-modal search spaces, as well as multi-models, multi-objective, constrained, and dynamic models [109]–[111]. While DE is not always the fastest approach, it is typically the one that delivers the best results, while the number of occasions where it is also quicker is substantial. DE also demonstrates its robustness regarding how control parameters are set and the consistency with which it discovers the genuine optimum. Furthermore, compared to one-point optimizers such as Powell's approach, DE is generally resistant to changes in beginning populations. DE is versatile enough to tackle situations when the objective functions lack the analytical definition required to compute gradients since it is a direct search approach. DE is also relatively simple to implement and change. DE is also easy to set up and tweak. The researchers discovered it is a suitable initial choice for starting a novel, complex global optimization problem with continuous and discrete parameters[69]. DE also can break free from local minima. DE has been used as a single-objective and multi-objective optimization approach to solve various engineering design challenges.

DE algorithms outperform Adaptive Simulated Annealing, the Annealed Nelder and Mead approach, GA, the Breeder GA, the easy evolution strategy, and the method of stochastic differential equations, as well as Particle Swarm Optimization algorithms, in terms of the required number of function evaluations necessary for locating a global minimum of the test functions [9], [111]–[113]. That is why we consider DE a global optimization tool to achieve the objective of our study.

2.4.5.1 DE structure and parameters:

DE algorithm is among the most successful evolutionary algorithms, simple but effective, that has demonstrated its ability to tackle many optimization issues. Since its inception, it has piqued the interest of numerous scholars who have proposed new, enhanced, state-of-the-art algorithms. DE operators and processes are presented in Figure 2-11; details are provided in the coming sections.

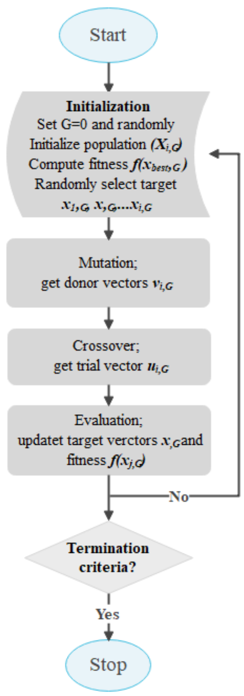

Figure 2-11: Differential Evolution (DE) Algorithm General Flow-Chart
Slika 2-11: Splošni diagram poteka algoritma diferencialne evolucije (DE)

A. Initialization:

The DE algorithm is a population-based meta-heuristic technique that uses a population of ***NP*** individuals, each represented by a vector of D-dimensional decision variables, as shown in Figure 2-11.

The population is randomly initiated within the search space ℝ, which includes all involved parameters' vectors, and $x^{r}_{i,G}$ denote each vector.

$$X_{i,G}^{j} = \left(x_{i,G}^{1}, x_{i,G}^{2}, x_{i,G}^{3}, \dots, x_{i,G}^{D}\right),\ i = 1,2,3,\dots,NP, j = 1,2,3,\dots,D, and\ G = 1,2,3,\dots,G_{max} \qquad 2.5$$

NP denotes the population size, ***D*** is the problem dimension, and ***G*** represents the generation or iteration number. Because decision variables are typically tied to physical components or measurements with natural boundaries, there may be a specified range in which the value of the decision variable should be defined for each specific problem. The Individuals inside the search space bounded by the specified minimum $\boldsymbol{x}_{min}$ and maximum $\boldsymbol{x}_{max}$ Constraints should be uniformly randomized to cover the whole range as much as possible in the initial population at the generation. $\boldsymbol{G}_0$.

An initialized individual vector can be defined as follows:

$$\boldsymbol{X}_{i,0}^{j} = \boldsymbol{x}_{min}^{j} + \boldsymbol{rand}.(\boldsymbol{x}_{max}^{j} - \boldsymbol{x}_{min}^{j}) \qquad 2.6$$

Where ***rand*** demonstrates a uniformly distributed random generator with a range of [0,1] for the $\boldsymbol{i}^{th}$ individual vector.

B. Mutation

Following the initiation stage, the mutation procedure produces a new offspring. The mutations of DE are characterized using the ***DE/x/y/z*** nomenclature. Where $\boldsymbol{x}$ represents the target vector, such as "random" or "best," $\boldsymbol{y}$ denotes the number of difference vectors used to modify $\boldsymbol{x}$, and $\boldsymbol{z}$ denotes the recombination operator employed, which might be binomial or exponential. By mutating each target vector $\boldsymbol{x}_{i,G}$, DE generates a donor vector $\boldsymbol{v}_{i,G}$, in the current iteration. Several generated mutation strategies are represented in the literature; the most often used mutation techniques are mentioned in [114]–[118].

C. Crossover (Recombination)

Crossover is another DE operator that follows the mutation phase to create a trial vector. $\boldsymbol{u}_{i,G}$ using target vector $\boldsymbol{x}_{r,G}$ and corresponding donor vector $\boldsymbol{v}_{i,G}$. To ensure population variety, the Crossover rate parameter Cr controls the amount of the perturbation of the base (target) vector. It was formerly assumed that crossover may exponentially enhance the chance of above-average parameter groups while lowering the probability of below-average groupings [77]. Research conducted more recently indicates that growth is not exponential since the selection advantage of a parameter grouping decreases as it becomes more prevalent. Furthermore, empirical data implies that (uniform) crossover does not reduce the computational complexity of an EA but rather speeds convergence by a constant factor. Despite this, crossover plays a substantial part in most EAs optimization systems[119].

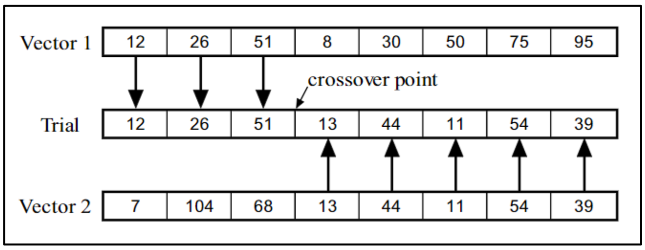

Figure 2-12: One-point Crossover example [107]
Slika 2-12: Primer enotočkovnega križanja

As illustrated in Figure 2-12, a vector of parameters is represented by each string. The trial vector receives a continuous series of parameter values from each vector. The crossing point is picked at random. It happens between the third and fourth factors in this situation. DE crossover procedures regulate the number of inherited components in a mutant vector to create a target vector. The most common crossover techniques are **Uniform** (binomial) and **exponential**.

a. Uniform (binomial)

Binomial or uniform crossover approaches are extensively used in the DE family of algorithms. Binomial crossover is implemented on each d-variable whenever a randomly generated value in the range [0,1] is less than or equal to a pre-determined value *Cr*, known as the crossover rate. In this scenario, the number of variables transmitted from the donor vector has a (near) binomial distribution. The scheme may be written as:

$$u_{i,G}^{j} = \begin{cases} v_{i,G+1}^{j} & if\ rand_{i,j}[0,1] \leq Cr \\ x_{i,G}^{j} & otherwise \end{cases} \qquad 2.7$$

Where $\boldsymbol{Cr}$ represents the crossover rate parameter, which usually ranges between 0 and 1, $\boldsymbol{rand}_{i,G}$ It is a uniform random integer that also ranges between 0 and 1. In each iteration, $\boldsymbol{rand}_{i,G}$ is instantiated once for each component of each vector and assures that the trial vector $\boldsymbol{u}_{i,G}$ receives at least one component from the target vector $\boldsymbol{v}_{i,G}$.

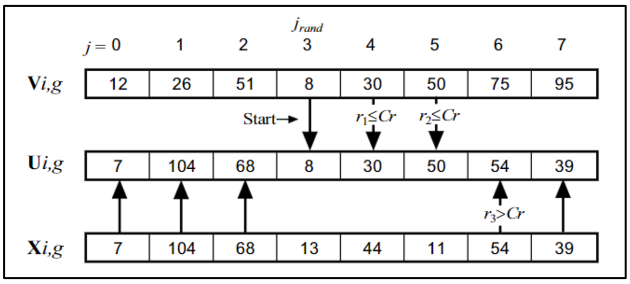

Figure 2-13: Uniform (Binomial) Crossover example [107]
Slika 2-13: Primer enotnega (binomskega) križanja

Figure 2-13 represents an example of binomial crossover, which could begin at any random place (e.g., $\boldsymbol{j}_{rand} = 3$) by producing a random number for each dimension; if the number is smaller than the crossover rate parameter $\boldsymbol{Cr}$, the trial vector is inherited from the mutant vector; otherwise, the target's vectors are copied.

b. Exponential

Although exponential crossover in DEs takes a different technique, it accomplishes the same result as one- and two-point crossovers. To make the trial vector $\boldsymbol{U}_{i,g}$ Distinct from the vector with which it will be compared, one parameter is picked randomly and duplicated from the mutant to the relevant trial parameter. Crossover rate *Cr* is compared to a uniformly distributed random number. (i.e., $\boldsymbol{rand}_j$ [0,1]) that is created new for each parameter to establish the source of the following trial parameters. Parameters are taken from the mutant vector as long as $(\boldsymbol{rand}_j\,[0,1] \leq Cr)$ is valid. Otherwise, the current and all remaining parameters are obtained from the target vector [107].

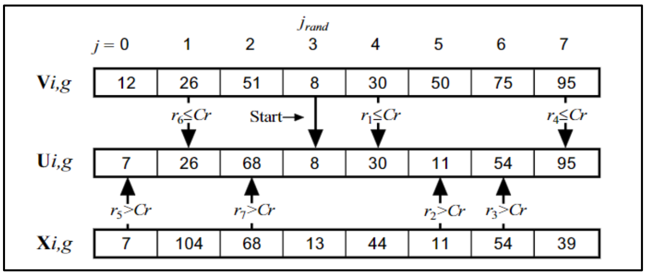

Figure 2-14: Exponential Crossover example [107]
Slika 2-14: Primer eksponentnega križanja

Figure 2-14 illustrates an example of an exponential crossover. It begins at any arbitrary point, such as $\boldsymbol{j}_{rand}$. It mutates each dimension of the trial vector. $\boldsymbol{U}_{i,g}$ until a random number bigger than ($\boldsymbol{Cr}$) is reached or the current individual's maximum dimension is reached. Because it ends if a single test condition is incorrect, this crossover scheme integrates less variety in the new person than the binomial crossover method.

D. Selection

A selection operation is used in the DE algorithm to create new population members. The selection operator employs a greedy approach, comparing the fitness of the trial vector. $\boldsymbol{u}_{i,G}$, to that of the target vector $\boldsymbol{x}_{r,G}$, and selecting the vector with the best fitness as a new population member, which replaces the target vectors for the next generation.

$$\boldsymbol{x}_{i,G+1} = \begin{cases} \boldsymbol{u}_{i,G} & if\ \boldsymbol{f}(\boldsymbol{u}_{i,G}) \leq \boldsymbol{f}(\boldsymbol{x}_{i,G}) \\ \boldsymbol{x}_{i,G} & otherwise \end{cases} \quad 2.8$$

Where the objective or cost function of the optimization problem is represented by ***f(.)***, as a result, if the new trial vector produces an equal or lower value of the objective function, it replaces the associated target vector in the next iteration; otherwise, the target is kept in the population of the following generation. Due to the crossover procedure, target and trial vectors can have the same numerical values for some decision variables. The three major parameters of the DE algorithms are mutation probability ($\boldsymbol{\mu F}$), crossover rate ($\boldsymbol{Cr}$), and population size (***NP***) [120], [121]. The mutation operator of the DE algorithm is performed to integrate new information into the population, whereas the crossover operator exchanges information between the trial and target vectors [122]. The parameter values used by the DE method are sensitive[116], [118], [123]. The control parameter ($\boldsymbol{\mu F}$) boosts convergence; with a small value, it focuses on exploitation, and with a considerable weight, it is on exploratory ability [123]. By rearranging competing vectors, the crossover strategy provides possible decomposability in the population (***NP***) [124], [125].

2.4.5.2 DE Literature Review

Several evolutionary algorithms (EAs) have been constructed as population-based strategies that address various optimization problems utilizing techniques derived from the natural evolution process. The reproduction operator is one of the primary distinctions amongst the established evolutionary algorithms[119]. The reproduction operator determines how new trial solutions are developed and evolved during optimization. The Differential Evolution (DE) approach is a stochastic metaheuristic algorithm that has successfully solved numerous optimization problems. The DE algorithm iteratively improves the given answer during the evolutionary search using mutation, crossover, and selection procedures. Sequential model-based algorithm control (SMAC) and multi-armed bandit optimization are two well-studied and generic methodologies for fine-tuning control settings [126], [127]. Many academics were inspired by this thought to create effective adaptive strategies for controlling parameters and process updates depending on various factors throughout the search. The algorithm's performance can be increased if the parameters and strategy adjustments are carefully thought out. Many scholars have used deterministic rules and adaptive/self-adaptive parameter control settings to regulate parameter settings for DE [128]. The first kind is a regulated parameter setup based on certain deterministic principles, with no information returned from the optimization search loop. The second method is primarily an adaptive control for the parameters based on the interplay of feedback information from the evolutionary search and control parameters. The central concept is to embed the parameter values into each individual and then develop them during the search. During evolutionary evolution, the values capable of producing superior offspring are more likely to survive to the following generation.

This category contains a large number of recently suggested DE algorithms. To improve the performance of their systems, several researchers combined the DE algorithm with other evolutionary algorithms or local search approaches. On the other hand, other techniques have used adaptive control settings and hybridization systems. Different improved DE methods have been developed in the literature during the last few years. Yang et al. [129] presented an auto-improved population diversity at the dimension level in 2015. The system measures the population distribution at each dimension to detect stagnation situations. If the algorithm detects stagnation during the search, the population is varied to an acceptable level, allowing the algorithm to achieve a higher convergence rate. To boost DE's performance, Li et al. [130] performed a fantastic job by employing two models; the first one is a distributed model that is used to develop new and better individuals and improve exploratory abilities, while the other approach is a centralized model for which the convergence speed was increased by using an evolution route based on covariance matrices adaption (CMA). When an individual gets stuck for several generations in an optimization system utilizing the DE algorithm, the parents are picked from an external archive that holds prior successful solutions [131]. Another enhancement to the DE algorithm is the addition of an adaptive ranking mutation operator, which ranks individuals based on three scenarios: infeasible, feasible, and semi-doable—this strategy aims to address the constrained issues [115]. Arithmetic recombination is combined with a DE technique that uses an ensemble of parameters to perform multi-modal optimization issues. The arithmetic recombination is used on the trailing vector with three random individuals with neighborhood mutation to enhance exploration and increase exploitation.

DE employs a primary mutation operator based on differences between pairs of solutions (referred to as vectors) to determine a search direction based on the distribution of solutions in the current population. DE also employs a steady-state-like replacement process, in which the freshly formed offspring (called a trial vector) competes exclusively against its matching parent (old object vector) and replaces it if the offspring has a better fitness value. DE has various similarities and differences with prior EAs. The following are examples of parallels: DE is a population-based technique in which crossover and mutation are the variation operators used to develop novel solutions, and a replacement mechanism gives the ability to keep the population size stable.

In contrast to GA, which might employ binary encoding, solutions in DE are coded using fundamental values. DE, on the other hand, does not employ a fixed distribution to regulate the behavior of the mutation operator; instead, the current distribution of the solutions in the search space defines the step size and search direction for each individual; this aspect appears to be one of its primary benefits [132].

3 RESEARCH METHODOLOGY

In the case of LUTI models, calibration generally entails the whole model design process, including establishing economic sectors, acquiring data, and zoning the research region. In this thesis, calibration refers to the method of estimating model parameters. The calibration procedure entails altering model parameters to duplicate data from a previous year in the research region. Getting a decent calibration is a time-consuming operation customarily done by experts and might take months. The different parameters of the model are estimated using a variety of approaches. Experts employ econometrical, ad-hoc processes and interactive trial-and-error methods to acquire results.

3.1 TRANUS

Consider a scenario where a region is divided into (*N*) sectors and (M) zones, and observed production and pricing statistics are available for a specific base year. The set of actual production and prices are denoted by $\boldsymbol{X}^{\text{act}}$ and $\boldsymbol{P}^{\text{act}} \in \mathbb{R}^{\text{N*M}}$, respectively.

The iteration process is designed in such a manner that the shadow prices ($\boldsymbol{h}$) are updated to push the productions ($\boldsymbol{X}$) to replicate the actual productions ($\boldsymbol{X}^{\text{act}}$) in the research region $\mathbf{R}^{\mathbf{N*M}}$. These variables will attempt to compensate for the other factors to achieve a perfect fit; these parameters serve as correction terms for parts of the utility that the model does not reflect.

The following equations (full mathematical description can be found here [18]) are governed by the combination of geographical zones ($\boldsymbol{i}$ & $\boldsymbol{j}$) and economic sectors ($\boldsymbol{n}$, $\boldsymbol{m}$ & $\boldsymbol{k}$) in a specific iteration (t).

The following equations are coded in Python-3 to replicate the TRANUS model. Further, this model will be called through the calibration approach.

Before the start of the iterations, the attractors for the induced production are calculated as follows:

$$\boldsymbol{A}_i^{nt} = \left(\textstyle\sum_k \boldsymbol{b}_k^n (\widetilde{\boldsymbol{X}}_i^{k,t-1})\right)\boldsymbol{W}_i^{n,t} \tag{3.1}$$

Where $\tilde{X}_i^{k,t-1}$ represent the total production (exogenous + induced), b_k^n relative weight and $W_i^{n,t}$ Initial attractor.

The iterations start with the computing of the induced demand, the first intermediate demand based on the following equations:

$$\boldsymbol{a}_i^{mn} = min^{mn} + (max^{mn} - min^{mn})exp\,(-\boldsymbol{\delta}^{mn}\boldsymbol{U}_i^n) \tag{3.2}$$

Where $min^{mn}\ and\ max^{mn}$ Represent the minimum and maximum amount of sector ($\boldsymbol{n}$) demanded by the unit production of sector $\boldsymbol{m}$, $-\boldsymbol{\delta}^{mn}$ represent the elasticity parameters and $\boldsymbol{U}_i^{mn}$ The disutility of consumptions for a sector ($\boldsymbol{n}$) in a zone($\boldsymbol{i}$). Then, here, the total demand for inputs ($\boldsymbol{n}$) in a specific area ($\boldsymbol{i}$) is calculated as follows:

$$\boldsymbol{D}_i^{mn} = (\boldsymbol{X}_i^{*m} + \boldsymbol{X}_i^m)\boldsymbol{a}_i^{mn}\boldsymbol{S}_i^{mn} \tag{3.3}$$

$$\boldsymbol{D}_i^n = \sum_m \boldsymbol{D}_i^{mn} + \boldsymbol{D}_i^{*n} \quad 3.4$$

Where $\boldsymbol{X}_i^{*m}$Represent the exogenous production, $\boldsymbol{X}_i^m$ the induced productions, $\boldsymbol{D}_i^{*n}$ exogenous demand and $\boldsymbol{D}_i^n$ Is the total demand for a sector ($\boldsymbol{n}$) in a zone ($\boldsymbol{i}$). $\boldsymbol{S}_i^{mn} = 1$, represents the substitution proportion of sector ($\boldsymbol{n}$) when consumed by sector ($\boldsymbol{m}$) in the zone ($\boldsymbol{i}$).

Production cost is calculated based on the consumption cost for the inputs to produce a sector (m) unit in a zone ($\boldsymbol{i}$).

$$\boldsymbol{C}_i^m = \left(\sum_n \boldsymbol{D}_i^{mn} \widetilde{\boldsymbol{C}}_i^n\right) + \boldsymbol{VA}_i^m \quad 3.5$$

Where $\widetilde{\boldsymbol{C}}_i^m$ Represent the consumption cost on input in ($\boldsymbol{n}$) in a zone ($\boldsymbol{i}$) and VA_i^m is the value added to the production of the sector ($\boldsymbol{m}$).

With the following equation, the utility for all productions and consumption will be calculated for the pairs of a zone ($\boldsymbol{ij}$) and sector ($\boldsymbol{n}$):

$$\boldsymbol{U}_{ij}^n = \boldsymbol{\lambda}^n\left(\boldsymbol{p}_j^n + \boldsymbol{h}_j^n\right) + \boldsymbol{t}_{ij}^n \quad 3.6$$

Where $\boldsymbol{p}_j^n$ represent the prices, $\boldsymbol{h}_j^n$ shadow price (adjustment parameters), $\boldsymbol{t}_{ij}^n$ transport disutility, and $\boldsymbol{\lambda}^n$ factor interpret the importance of the prices.

Once the utilities are computed, the probability that the sector ($\boldsymbol{n}$) 's production, which is needed in zone ($\boldsymbol{i}$) , is in zone ($\boldsymbol{j}$), can be calculated as follows.

$$\boldsymbol{Pr}_{ij}^n = \frac{\boldsymbol{A}_j^n exp\left(-\boldsymbol{\beta}^n \boldsymbol{U}_{ij}^n\right)}{\sum \boldsymbol{A}_j^n \exp\left(-\boldsymbol{\beta}^n \boldsymbol{U}_{ij}^n\right)} \quad 3.7$$

Where $\boldsymbol{A}_j^n$ denotes the production attractor, $\boldsymbol{\beta}^n$ dispersion parameter, and $\boldsymbol{U}_{ij}^n$utilities.

With the $\boldsymbol{Pr}_{ij}^n$ and total demand $\boldsymbol{D}_i^n$ total induced production of sector ($\boldsymbol{n}$), which is assigned to zone ($\boldsymbol{i}$), will be calculated as follows:

$$\boldsymbol{X}_{ij}^n = \boldsymbol{D}_i^n \boldsymbol{Pr}_{ij}^n \quad 3.8$$

$$\boldsymbol{X}_i^n = \sum_i \boldsymbol{X}_{ij}^n \quad 3.9$$

Where $\boldsymbol{X}_{ij}^n$ production for the combination of the sector ($\boldsymbol{n}$), production zone ($\boldsymbol{j}$), and consumption zone ($\boldsymbol{i}$).

After the demand is assigned to each production zone, the consumption cost for each unit input sector ($\boldsymbol{n}$) is computed.

$$\widetilde{\boldsymbol{C}}_i^n = \frac{\sum_j \boldsymbol{X}_{ij}^n\left(\boldsymbol{p}_j^n + \boldsymbol{tm}_{ij}^n\right)}{\sum_j \boldsymbol{X}_{ij}^n} \quad 3.10$$

Where $\boldsymbol{tm}_{ij}^n$ represent the monetary cost needed to transport a unit of ($\boldsymbol{n}$) from the production zone ($\boldsymbol{j}$) to the consumption zone ($\boldsymbol{i}$).

At the end of each iteration, the shadow prices are computed as follows.

$$\boldsymbol{h}_{i,G+1}^n = \left(\boldsymbol{h}_{i,G}^n + \boldsymbol{P}_{i,G}^n\right)\frac{\boldsymbol{X}_{i,G}^n}{\boldsymbol{X}_{act,G}^n} - \boldsymbol{P}_{i,G+1}^n \quad 3.11$$

Where shadow prices are presented by $\boldsymbol{h}_{i,G}^{n}$, $\boldsymbol{X}_{act,G}^{n}$ denoted the actual (observed) production, model production, and prices are represented by $\boldsymbol{X}_{i,G}^{n}$ and $\boldsymbol{P}_{i,G}^{n}$, economic sectors, geographic zones, and generation (iteration) numbers are denoted with $\boldsymbol{n}$, $\boldsymbol{i}$, and $\boldsymbol{G}$**,** respectively.

3.2 Proposed calibration approach

A fully automatic and global calibration approach for the LUTI models has been a wish for all the experts in recent years. Toward this goal, in this thesis, a novel method using the Differential Evolution algorithm is proposed and tested on the TRANUS land-use module. The purpose is to replace the TRANUS current sequential calibration procedure with an automatic and global estimation approach using a multi-objective optimization technique so that any parameters can be calibrated according to their restrictions.

We argue that using a differential evolution algorithm as the calibration technique and measuring the model performance using multi-objective functions MANE and RMSE concerning a set of constraints is a natural method to achieve this goal. The present TRANUS calibration technique has a scenario where shadow prices and other parameters are estimated without using an objective function.

Further sections, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) were utilized for evaluation purposes. Ultimately, optimizing performance improvement is implemented using Hybrid PSODE and Hybrid GADE.

3.2.1 Sensitivity Analysis

For accurate model tuning and superior prediction abilities, evaluating the sensitivity of the input parameters on the outputs during the calibration phase is essential. The dimension of the entire calibration problem may also be significantly reduced with such an analysis if many parameters are involved, facilitating parameter estimation. Therefore, the first step of the proposed calibration approach is to conduct a sensitivity analysis of the desired parameters to identify which input parameters affect the variation of the outputs the most. Then, these parameters and shadow prices are calibrated; the methodology for this calibration approach is discussed in the subsequent section.

Sensitivity analysis refers to how variation in the output of a numerical model can be attributed to variations in its input. Assessing the sensitivity of the input parameters on the results is a crucial step to properly calibrating the model and ensuring better predicting capabilities. To calibrate LUTI models, an essential number of parameters must be estimated, which will be time-consuming depending on the estimation technique used.

Global sensitivity analysis is used for a range of very diverse purposes, such as supporting model calibration, verification, diagnostic evaluation [133], [134], prioritizing efforts for uncertainty reduction [135], analyzing the dominant controls of a system [136], and to support robust decision-making [137].

The first sensitivity analysis was conducted on TRANUS by Dutta et al. [17], using "pick-freeze" estimation techniques. Among other available approaches, the variance-based method allows calculating sensitivity indices called Sobol' indices [139]. The influence of each input or set of information is represented by these indices, which range between 0 and 1 –the higher the index, the more influential the input. Higher-order indices estimate the equivalent relevance of interactions between information, while first-order indices estimate the principal influence from each input. Various estimating methodologies have been utilized in the literature [140] to calculate Sobol indices.

In our study, a sensitivity analysis has been conducted using generalized Sobol indices [139] on the land-use and activity module parameters of the TRANUS, which are presented in Table 3-1.

Table 3-1: Parameters that are assumed to be unknown
Preglednica 3-1: Parametri, za katere se domneva, da so neznani

Parameters	Description	Involved Equation
$\boldsymbol{\delta}^{mn}$	elasticity parameter of the sector (m) concerning the error of sector(n)	3.2
$\boldsymbol{b}^{n}{}_{k}$	the relative weight of sector (k) as an attractor to sector (n)	3.1
$\boldsymbol{\beta}^{n}$	dispersion parameter of multinomial logit model for sector (n)	3.7
$\boldsymbol{W}^{n}{}_{j}$	the initial attractor of the zone (j) considering non-modeled elements that attract the location of the sector (n)	3.1
$Atrac.Fac^{n}$	attractor factor concerning sector (n)	3.7
$\boldsymbol{\lambda}^{n}$	the factor that regulates the relative importance of prices versus transport disutility in the utility function related to the sector (n)	3.6

The TRANUS model was first coded using Python to apply the sensitivity analysis. Then, the generalized Sobol indices for land-use and activity module parameters that were assumed unknown were estimated using an open-source Python library known as SALIB [141]. In this process, Mean Absolute Normalized Error (MANE) has been used as an error function, and the influences of input values on the MANE values of the productions ($\boldsymbol{X}^{n}{}_{j}$) and prices ($\boldsymbol{P}^{n}{}_{j}$) that act as the TRANUS land use and activity of model output were calculated.

The MANE formula is given below:

$$Oput_{MANE} = \boldsymbol{MANE}_{X} + \boldsymbol{MANE}_{P} = \frac{1}{N}\sum_{1}^{N}\left|\frac{\boldsymbol{X}_{act}-\boldsymbol{X}_{mod}}{\boldsymbol{X}_{act}}\right| + \frac{1}{N}\sum_{1}^{N}\left|\frac{\boldsymbol{P}_{act}-\boldsymbol{P}_{mod}}{\boldsymbol{P}_{act}}\right| \tag{3.12}$$

Whereas $\boldsymbol{X}_{act}$ and $\boldsymbol{P}_{act}$ are observed productions and prices, $\boldsymbol{X}_{mod}$ and $\boldsymbol{P}_{mod}$ are modeled productions and prices, $\boldsymbol{N}$ is the number of observations, and $\boldsymbol{Oput}_{MANE}$ acts as the model overall output values against input parameters given in Table 3-1.

Figure 3-1 and Figure 3-2 show the sensitivity analysis results carried out on five sets of TRANUS land use and activity model parameters.

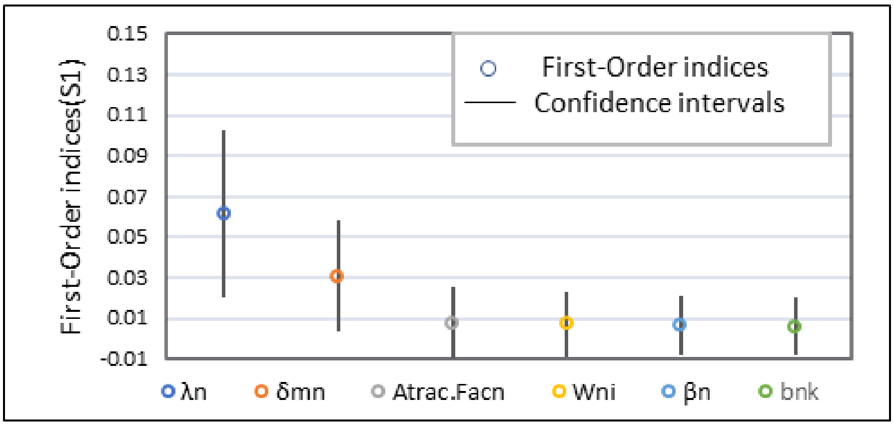

Figure 3-1: Sobol indices estimation for the parameters of Table 3-1 (Average First-order)
Slika 3-1: Ocena Sobolovih indeksov za parametre Table 3-1 (Povprečje prvega reda)

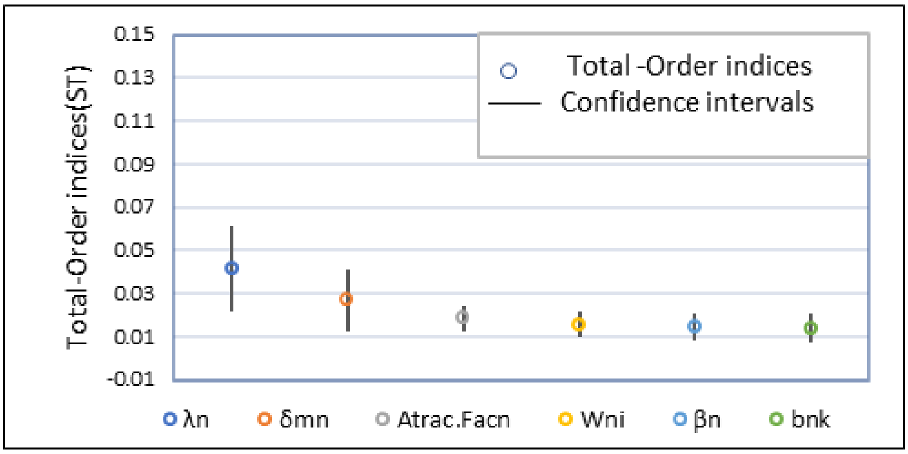

Figure 3-2: Sobol indices estimation for the parameters of Table 3-1 (Average Total-order)
Slika 3-2: Ocena Sobolovih indeksov za parametre Table 3-1 (povprečni skupni vrstni red)

Sensitivity analysis results show that price factor (λ^n) and elasticity (δ^{mn}) have the highest impact on model output (***Oput***$_{MANE}$). Therefore, these parameters are selected to be used in the calibration process alongside shadow prices (h^n_j) and play a significant role (adjustment factor) in the calibration of the TRANUS land-use and activity model, as they represent attributes of the socio-economic system that are not included in the model.

3.2.2 Objective (Cost) Functions

An optimization problem's ultimate purpose is to improve the objective function(s), which can be described by a single or possibly multiple expressions. Most optimization methods involve a single objective or cost function that reduces a cost or time or maximizes a specific value or income. Multi-objective functions are used to discover the optimum trade-off between two or more associated problem objectives.

The Root Mean Square Error (RMSE) and the Mean Absolute Normalized Error (MANE) are two multi-objective functions used to calibrate model parameters in prior research and are extensively utilized worldwide. In several studies and research investigations (i.e., meteorology, air quality, and climate), RMSE and MANE have been employed as standard statistical tools to quantify model performance [142]–[144]. MANE and RMSE are also implemented as multi-objective functions in many traffic and transportation model calibration studies and proved their effectiveness as indicators for model assessments [145]–[152].

Even though they have been used to evaluate model performance for many years, there is no agreement on the best metric for model error.

In this thesis, these two multi-objective functions (MANE and RMSE) are utilized to minimize the error between actual and simulated results of productions ($\boldsymbol{X^n_j}$) and prices ($\boldsymbol{P^n_j}$). To this end, the difference between actual and simulated productions and prices are formalized in the following equations, where f_1 and f_2 are the RMSE values and f_4 and f_5 are MANE values of production and prices, respectively. RMSE and MANE values of production and prices are linearly scalarized with the same weight in f_3 and f_6, respectively. Both f_3 and f_6 are the model objective (cost) functions that must be minimized as much as possible during calibration.

$$f_1(x) = RMSE_X = \sqrt{\frac{\sum_1^N (\boldsymbol{X}_{act} - \boldsymbol{X}_{mod})^2}{\boldsymbol{N}}} \tag{3.13}$$

$$f_2(x) = RMSE_P = \sqrt{\frac{\sum_1^N (\boldsymbol{P}_{act} - \boldsymbol{P}_{mod})^2}{\boldsymbol{N}}} \tag{3.14}$$

$$f_3(x) = RMSE_X + RMSE_P \tag{3.15}$$

$$f_4(x) = MANE_X = \frac{1}{N} \sum_1^N \frac{|\boldsymbol{X}_{act} - \boldsymbol{X}_{mod}|}{X_{act}} \tag{3.16}$$

$$f_5(x) = MANE_P = \frac{1}{N} \sum_1^N \frac{|\boldsymbol{P}_{act} - \boldsymbol{P}_{mod}|}{\boldsymbol{P}_{act}} \tag{3.17}$$

$$f_6(x) = MANE_X + MANE_P \tag{3.18}$$

Xact and Pact are observed productions and prices, Xmod and Pmod are modeled productions and prices, N represents the number of observations, and f(x) is the objective function.

3.2.3 Calibration and optimization techniques

LUTI models have attracted the attention of researchers in recent years. They are interested in developing models which are user-friendly, generic, and have trustable results. Over the past years, many LUTI models have been developed and applied to different regions and cities. It is a fact that, in developing land use and transportation integration models, a set of complex nonlinear systems is employed. Analysis of such systems is a complex and time-consuming task, especially with uncertainty. Calibration of LUTI models plays a key role by determining the optimal parameters and creating a trustable model, for the decision maker.

Several optimization methods have been employed during the models' parameters calibration, including Genetic Algorithm (GA), [86], [153], [154] Particle Swarm Optimization (PSO) [39], [99], Maximum-Likelihood Estimation [55], Differential Evolution (DE) algorithm[107], [132], [155]–[157]. Although many more optimization techniques are utilized in the calibration process, here in this thesis, the DE algorithm is selected to be used through the proposed calibration approach because of the many advantages mentioned in previous sections.

3.2.4 Differential Evolution Algorithm

DE algorithm is widely used due to its simplicity and ease of implementation. DE algorithm's population of possible solutions is randomly initiated inside an n-dimensional search space where all candidate solutions have an equal chance of being chosen as parents. Candidate solutions emerge by examining the whole search space throughout the cycles to find the objective function's optimum. The DE algorithm employs four fundamental processes: **initialization**, **mutation**, **crossover**, and **selection**, as they are common in other EA optimization strategies but have slightly different principles.

The pseudocode of the DE algorithm, which is obtained from [158], is adapted to our study example for the LUTI model calibration purposes, as shown in Algorithm 3-1. This algorithm consists of two main parts: initialization and the main loop. The objective function and DE parameters' values were defined in the initialization part, and then a random initial population was generated. Furthermore, the TRANUS model was run, and the objective function for each member of the population was determined to obtain the optimal objective function value and its matching population. Then, in the main loop, for each member of the population in every iteration, the TRANUS model was run to evaluate the objective function (or cost function).

In the mutation stage, each parameter's lower and upper bound values are generally used to clip mutated values. In contrast, in this proposed approach, a specific technique is used to clip the mutant values, where the values for each parameter obtained in the mutation process were evaluated with the upper and lower bounds of the desired parameters (shadow prices, lambda, and elasticity), if the mutation value obtained was out of the bound, the value of the current population was replaced as the mutant value.

Algorithm 3-1: Calibration of land use model parameters using modified DE algorithm

Initialization

Define Cost Functions [MANE and RMSE]

Generate a random initial population $\{x_{i,0} \mid i = 1,2, \ldots, NP\}$

Evaluate Objective function,

$$\boldsymbol{fitness = \{f(x_{i,G}), i = 1, 2, \ldots, Np\}} \quad \%\% \boldsymbol{RUN\ TRANUS}$$

$best_{index} = \arg\min\ (fitness)$

$best = x_{best,G}$

Main Loop

For i in range (*MaxIter*):

For j in range (*NP*):

For k in range (*NP*):

Select randomly

$$x_{r1,G}, x_{r2,G}, x_{r3,G}, x_{r4,G}, x_{r5,G} \in [1; Np, replace = False]$$

#Mutation (generate donor vector)

$$v_{k,G} = x_{r1,G} + \mu_{F1}.(x_{r2,G} - x_{r3,G}) + \mu_{F2}.(x_{r4,G} - x_{r5,G})$$

End For

#Crossover (generate trail vector)

Generate randomly, $CR_{rand[0,1]} = randint[1; k]$

If $CR_{rand[0,1]} < \mu_{CR}$

$$u_{j,G} = v_{j,G}$$

Else

$$u_{j,G} = x_{j,G}$$

End If

Evaluation of objective function,

$$\boldsymbol{fitness = \{f(u_{j,G}), i = 1, 2, \ldots, Np\}} \quad \%\% \boldsymbol{RUN\ TRANUS}$$

#Selection

If $f(u_{j,G}) < f(x_{j,G})$,

$x_{j,G} = u_{j,G}$

$f(x_{j,G}) = f(u_{j,G})$

If $f(u_{j,G}) < f(x_{best,G})$

$best_{index} = j$

$x_{best,G} = u_{j,G}$

End If

End If

End For

If the repetition of $f(x_{best,G}) = 40\ OR$

$(f(x_{best,G}) - AVE(f(u_{j,G}))) < 0.000001$

BREAK

End For

3.2.4.1 Parameter Settings

Using the DE algorithm as an optimization technique has varying performance depending on the mutation strategy and selection of control parameters. For different optimization problems, the most appropriate mutation method and parameter settings vary. When addressing a particular topic, the parameter values for any optimization approach should be set beforehand. Reviews of the literature reveal that no single parameter setting works for all kinds of issues.

Control parameters may significantly influence the performance of any evolutionary algorithm, and the same is true for the DE algorithm. After being specified, the control parameters are fixed during the search process. In [159], it is proposed that mutation factor (µF) and crossover rate (CR) should have values between [0.4, 1] and [0.5, 0.7], respectively, and that NP should be [240] *D, where D is the problem dimension. Different control parameters have various effects on the algorithmic performance in terms of effectiveness, efficiency, and resilience. However, a range of values is used in previous studies for each parameter setting involved in the DE algorithm. It is claimed that NP ∈ [3, 8] *D, μ_F= 0.6 and CR ∈ [0.3, 0.9] are the most appropriate initial selections [160], while Population size NP ∈ [3, 10] *D and mutation factor µF ∈ [0.5, 1.0], are chosen in most cases [9]. Different kinds of parameter-setting selections can be found in most cases in the literature, so to have the best selections, here in this thesis, the problem is programmed automatically to search for the best combinations of the parameter's values in the specific range presented in Table 3-2. The initial potential solutions are mutated using various strategies and parameter factors. Several types of mutation strategies are used in DE algorithm applications for different types of problems, while the most common ones are introduced in the literature review (section 2.4.5). Each DE strategy has its advantages and disadvantages depending on the application purpose. Improvement of the mutation strategy is essential to getting the best out of the DE algorithm. Several generated mutation strategies are represented in the literature; the five most often used mutation techniques are modified according to the example used in this thesis. A significant improvement is applied to the mutation strategies (Equations 3.21, 3.22, 3.23), where two different mutation factors are present. $\boldsymbol{\mu_{F1}}$ and $\boldsymbol{\mu_{F2}}$ are used, instead of single factor. The results show a significant improvement in the modeled values of the parameters during the calibration process.

$$DE/rand/1 : \boldsymbol{v}_{i,G} = \boldsymbol{x}_{r1,G} + \boldsymbol{\mu_F}\cdot(\boldsymbol{x}_{r2,G} - \boldsymbol{x}_{r3,G}) \quad 3.19$$

$$DE/best/1 : \boldsymbol{v}_{i,G} = \boldsymbol{x}_{best,G} + \boldsymbol{\mu_F}\cdot(\boldsymbol{x}_{r1,G} - \boldsymbol{x}_{r2,G}) \quad 3.20$$

$$DE/current-to-best/1: \boldsymbol{v}_{i,G} = \boldsymbol{x}_{r1,G} + \boldsymbol{\mu_{F1}}\cdot(\boldsymbol{x}_{best,G} - \boldsymbol{x}_{current,G}) + \boldsymbol{\mu_{F2}}\cdot(\boldsymbol{x}_{r1,G} - \boldsymbol{x}_{r2,G}) \quad 3.21$$

$$DE/best/2 : \boldsymbol{v}_{i,G} = \boldsymbol{x}_{best,G} + \boldsymbol{\mu_{F1}}\cdot(\boldsymbol{x}_{r1,G} - \boldsymbol{x}_{r2,G}) + \boldsymbol{\mu_{F2}}\cdot(\boldsymbol{x}_{r3,G} - \boldsymbol{x}_{r4,G}) \quad 3.22$$

$$DE/rand/2 : \boldsymbol{v}_{i,G} = \boldsymbol{x}_{r1,G} + \boldsymbol{\mu_{F1}}\cdot(\boldsymbol{x}_{r2,G} - \boldsymbol{x}_{r3,G}) + \boldsymbol{\mu_{F2}}\cdot(\boldsymbol{x}_{r4,G} - \boldsymbol{x}_{r5,G}) \quad 3.23$$

Where i, marks the current population member, and G represents the number of generations.

Target vectors $\boldsymbol{x}_{r,G}$ which are randomly selected from the range $[1, NP]$ and indicted by random indices $[\boldsymbol{r}_1, \boldsymbol{r}_6]$ are differed from the current vector $\boldsymbol{x}_{i,G}$. The best individual vector, which indicates the best fitness value (i.e., for a minimization problem, it can be the lowest value of the cost function) in the whole population, is presented by $\boldsymbol{x}_{best,G}$. The difference vectors are scaled using the scaling factor. $\boldsymbol{\mu}_{\boldsymbol{F}\cdot}$, which is a positive control parameter. DE mutation strategies are implemented with automatic programming, using MANE and RMSE, in the ranges of values mentioned in Table 3-2.

Table 3-2: Parameters settings for DE operators (RMSE and MANE)
Preglednica 3-2: Nastavitve parametrov za operaterje DE (RMSE in MANE)

Parameters	Settings	Description
MaxIt	400 for MANE),1000 for RMSE	maximum number of generations
NP	[10, 100] In steps 5	initial population number
μ_{F1}	[0.1, 2.0] In steps 0.1	mutation factor
μ_{F2}	[0.1, 2.0] In steps 0.1	mutation factor
Cr	[0.1, 0.9] In steps 0.1	crossover probability

The best outcomes of all five mutation strategies using the mentioned ranges of values are presented in Table 3-3 and Table 3-4.

Table 3-3: DE mutation strategy evaluation results using MANE
Preglednica 3-3: Rezultati vrednotenja strategije mutacije DE z uporabo MANE

Parameters	***NP***	$\boldsymbol{\mu}_{\boldsymbol{F1}}$	$\boldsymbol{\mu}_{\boldsymbol{F2}}$	***Cr***
$DE/rand/1$	20	0.4	-	0.9
$DE/best/1$	20	2.0	-	0.7
$DE/current - to - best/1$	20	0.1	1.7	0.5
$DE/best/2$	20	0.7	2.0	0.1
$DE/rand/2$	20	0.2	0.2	0.8

Table 3-4: DE mutation strategy evaluation results using RMSE
Preglednica 3-4: Rezultati vrednotenja strategije mutacije DE z uporabo RMSE

Parameters	***NP***	$\boldsymbol{\mu}_{\boldsymbol{F1}}$	$\boldsymbol{\mu}_{\boldsymbol{F2}}$	***Cr***
$DE/rand/1$	30	0.5	-	0.9
$DE/best/1$	30	2.0	-	0.3
$DE/current - to - best/1$	30	0.3	1.9	0.8
$DE/best/2$	30	0.3	0.8	0.4
$DE/rand/2$	30	0.3	0.3	0.9

Table 3-5: Parameters selected for DE operators (RMSE and MANE)
Preglednica 3-5: Parametri, izbrani za operaterje DE (RMSE in MANE)

Parameters (RMSE)	Parameters (MANE)	Description
MaxIt = 1000	*MaxIt* = 400	maximum number of generations
NP = 30	*NP* = 20	initial population number
μ_{F1} = 0.3	μ_{F1} = 0.2	mutation factor
μ_{F2} = 0.3	μ_{F2} = 0.2	mutation factor
Cr = 0.9	*Cr* = 0.8	crossover probability

As illustrated in Figure 3-3 and Figure 3-4, the mutation schema $DE/rand/2$, (Equation 3.23) using the parameters listed in Table 3-5 had an outstanding performance with the. This mutation schema modeled production $\boldsymbol{X^n}_j$ and prices $\boldsymbol{P^n}_j$ almost identical to the actual or observed data, using multi-objective functions RMSE and MAN.

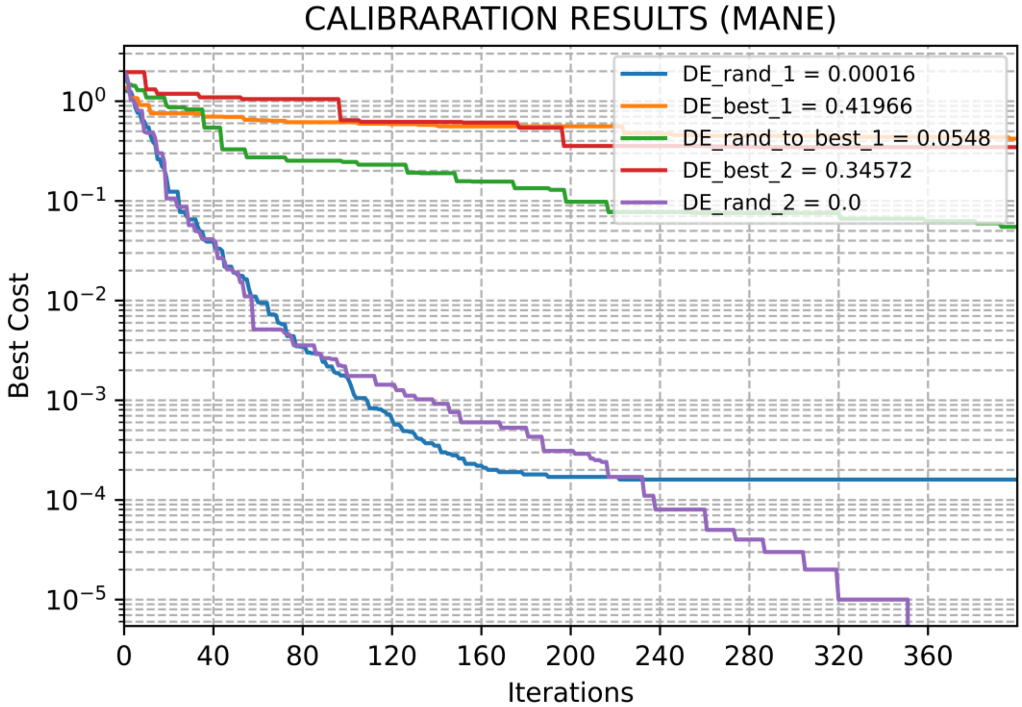

Figure 3-3: MANE values of DE mutation strategies

Slika 3-3: Vrednosti MANE strategij mutacije DE

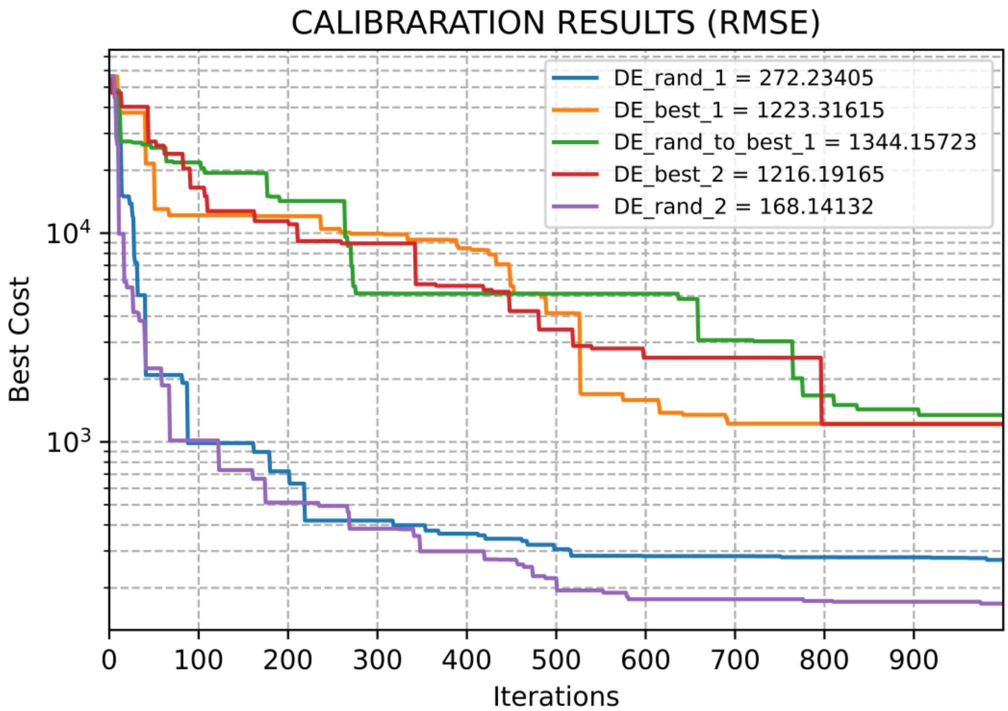

Figure 3-4: RMSE values of DE mutation strategies

Slika 3-4: RMSE vrednosti mutacijskih strategij DE

3.3 Genetic Algorithm

As described in previous sections, GA starts with a randomly selected population of individuals/candidates, which might be a possible solution to the problem. The solutions are evaluated using the fitness value of each candidate. Individual fitness represents the suitability of the selected solution. Then, parents are selected for the reproduction process based on their fitness value. With a cross-over probability, parents are combined to produce new offspring (children) using the uniform crossover techniques. Finally, individuals/gens selected based on mutation probability are modified and ranked according to their fitness through the evaluation process. Although several GA versions have been developed so far in our case, the GA version coded by Kamel [161] is adapted to our study example as presented in Algorithm 3-2.

Algorithm 3-2: Calibration of land use model parameters using modified GA algorithm

Initialization
Define Cost Functions [MANE and RMSE]
Generate a random initial population. $\{x_{i,0} \mid i = 1,2,\ldots,NP\}$
For i in range (*NP*)
#Evaluate Cost function,
$\boldsymbol{fitness = \{f(x_{i,G}), i = 1,2,\ldots,Np\}}$ %%$\boldsymbol{RUN\ TRANUS}$
End For
$best_{index} = arg\ min\ (fitness)$
$bestsol = x_{best}$

bestcost = np.empty(***MaxIt***)
Main Loop
For i in range (***MaxIt***):
Crossover Operation
For *j* in range (*nCrossover*)
Select parents randomly
Generate *offspring*
#Evaluate cost functions
Cost Evaluation ($\boldsymbol{X^n{}_j}$, $\boldsymbol{P^n{}_j}$) (%%$\boldsymbol{RUN\ TRANUS}$)
Update $bestsol$
End For
#Mutation Operation
For *j* in range (*nMutation*)
Select parents randomly
Create *offspring* using parents
#Evaluate cost functions
Cost Evaluation ($\boldsymbol{X^n{}_j}$, $\boldsymbol{P^n{}_j}$) (%%$\boldsymbol{RUN\ TRANUS}$)
Update $bestsol$
#Merge, Sort, and Selection
Population merging
Population sorting
Exclude extra population
Generate new population
#Update best cost
bestcost[*it*] = *bestsol*
End For

3.3.1 Parameter Settings

GA design is limited to balancing crossover and mutation rates [162]. Thus, GA is evaluated over a general range of crossover probability (***Cr***) and mutation rates (***mu***) presented in Table 3-6 for the example used in this thesis. The best combinations of the GA operators' values are selected through an automatic process for RMSE and MANE, listed in Table 3-7.

Table 3-6 Parameters settings for GA operators (RMSE and MANE)
Preglednica 3-6: Nastavitve parametrov za operaterje GA (RMSE in MANE)

Parameters	Settings	Description
MaxIt	400 for MANE),1000 for RMSE	maximum number of generations
NP	[10, 100] In steps 5	initial population number
Cr	[0.1, 1.0] In steps 0.1	crossover probability
mu	[0.1, 1.0] In steps 0.1	mutation rate

Table 3-7: Parameters selected for GA operators (RMSE and MANE)
Preglednica 3-7: Parametri, izbrani za operaterje GA (RMSE in MANE)

Parameters (RMSE)	Parameters (MANE)	Description
MaxIter = 1000	*MaxIter* = 400	maximum number of generations
NP = 30	*NP* = 20	initial population number
Cr = 1	*Cr* = 1	crossover probability
mu = 0.18	*mu* = 0.18	mutation rate
sigma = 0.1	*sigma* = 0.1	mutation step size

3.4 Particle Swarm Optimization

PSO is a population-based algorithm requiring two elements: search space (a swarm of particles) and particles (potential solutions). As in DE and GA, PSO also starts with initialization, where a particle swarm will be generated randomly based on defined parameters and proceeds by calculating the objective function depending on the position and velocity of each member (particle). Then, the objective function values are compared to the global objective function values to see which is superior. The best particle information calculates the new particle velocity and location. The fundamental benefit of PSO is that at every iteration, information flows between all particles. The particles rely on other data to arrive at the optimal answer. In our case, we used the PSO version coded by Kamel [161], and it is adapted to our study example as presented in Algorithm 3-3.

Algorithm 3-3: Calibration of land use model parameters using modified PSO algorithm

```
Define Cost Functions [MANE and RMSE]
Set Bounds [VarMin, VarMax]
Gbest = {position: None, cost: np.inf}
PSO Initialization Loop
    For i in range (NP)
        Initialize the position and velocity of particles randomly
        Calculate cost of particles (X^n_j, P^n_j) (%%RUN TRANUS)
        #Update best personal [position, velocity]
        If Pop[i][cost] < Pop[i][bestcost]
            Pop[i][bestposition]=Pop[i][position];
            Pop[i][bestcost]=Pop[i][cost];
        End If
    #Update best Global
        If Pop[i][bestcost] < Gbest[cost]
            Gbest[position]= Pop[i][bestposition]
            Gbest[cost]= Pop[i][bestcost]
        End If
    End For
#Initialize best cost
bestcost = np.empty(MaxIter)
PSO Main Loop
    For it in range (MaxIter) %%Stopping criteria)
        For i in range (NP)
            #Update particles velocity
            Pop[i][velocity]
                = Pop[i][velocity] + c_1 * rand
                * (Pop[i][bestposition] - Pop[i][position]) + c_2
                * rand * (Gbest[position] - Pop[i][position])
            Apply Bounds
        Cost Evaluation (X^n_j, P^n_j) (%%RUN TRANUS)
            #Update best personal using
        If Pop[i][cost] < Pop[i][bestcost]
            Pop[i][bestposition] = Pop[i][position];
            Pop[i][bestcost] = Pop[i][cost];

            #Update best Global
            If   Pop[i][best] < Gbest[cost]
                 Gbest[position] = Pop[i][best]
                 Gbest[cost] = Pop[i][bestcost]
            End If
        End If
    End For
    Return bestcost[it] = Gbest[cost]
```

3.4.1 Parameter Settings

PSO parameters, personal learning coefficient (***C**$_1$*), population (global) learning coefficient (***C**$_2$*), and inertia weight (***w***) are evaluated over a range of values, as mentioned in Table 3-8. The best combinations are selected automatically for RMSE and MAN, as presented in Table 3-9.

Table 3-8: Parameters settings ranges for PSO operators (RMSE and MANE)
Preglednica 3-8: Območja nastavitev parametrov za operaterje PSO (RMSE in MANE)

Parameters	Settings	Description
MaxIt	400 for MANE),1000 for RMSE	maximum number of generations
NP	[10, 100] In steps 5	initial population number
C_1	[0.1, 2.0] In steps 0.1	personal learning coefficient
C_2	[0.1, 2.0] In steps 0.1	global acceleration coefficient
w	[0.1, 1.0] In steps 0.1	inertia weight

Table 3-9: Parameters selected for PSO operators (RMSE and MANE)
Preglednica 3-9: Parametri, izbrani za operaterje PSO (RMSE in MANE)

Parameters (RMSE)	Parameters (MANE)	Description
MaxIt = 1000	*MaxIt* = 400	maximum number of generations
NP = 30	*NP* = 20	initial population number
C_1 = 1.5	C_1 =1.2	personal acceleration coefficient
C_2 = 1.3	C_2 = 1.2	global acceleration coefficient
w = 0.9	w = 0.9	inertia weight

3.5 HYBRID Strategy

Combining the operators and varieties of different optimization techniques is known as Hybridization. It is one of the most effective and efficient strategies to enhance the performance of optimization strategies. Hybridizations replace some algorithms' weaknesses, such as convergence speed, low accuracy, and sticking to a solution with the advantages of the other algorithms. In several studies, hybrid algorithms were tested and showed their effectiveness in solving the most complex optimization problems [114], [163]–[166].

This study tried to improve the performance of the proposed calibration approach with the two well-known Hybrid algorithms, PSODE and GADE. In the meantime, utilizing these algorithms further evaluates the calibration approach proposed and developed in this thesis.

3.5.1 Hybrid PSODE algorithm

PSO and DE are stochastic algorithms far more effective than the gradient descent approach for achieving the global optimum. On the other hand, by closely examining each of them, some strengths and weaknesses come out. A hybridization of PSO and DE is implemented to enhance optimization performance by addressing these limitations and using the advantages of each one. Merging can be done in each or one of the leading operators of the algorithms, initialization, perturbation, or evaluation and selection phase, whether it is PSODE (PSO as the base and DE operators are merged in it) or DEPSO (DE as the base and PSO operators is merged in it).

In this thesis, the PSODE algorithm is utilized in the study example. The version used here is inspired by the versions presented in [167] and [168].

Algorithm 3-4: Calibration of land use model parameters using modified PSODE algorithm

```
Define Cost Functions [MANE and RMSE]
Set Bounds [VarMin, VarMax]
Gbest = {position: None, cost: np.inf}
PSO Initialization Loop
    For i in range (NP)
        Initialize the position and velocity of particles randomly
        Calculate cost of particles (X^n_j, P^n_j) (%%RUN TRANUS)
        #Update best personal [position, velocity]
        If Pop[i][cost] < Pop[i][bestcost]
            Pop[i][bestposition]=Pop[i][position];
            Pop[i][bestcost]=Pop[i][cost];
        End If
        #Update best Global
        If Pop[i][bestcost] < Gbest[cost]
            Gbest[position]= Pop[i][bestposition]
            Gbest[cost]= Pop[i][bestcost]
        End If
    End For
#Initialize best cost
bestcost = np.empty(MaxIt)
PSO Main Loop
    For it in range (MaxIt) %%Stopping criteria)
        For i in range (NP)
            #Start DE
            Select randomly
                p_{r1,G}, p_{r2,G}, p_{r3,G}, p_{r4,G}, p_{r5,G}, ∈ [1; Np, replace = False]
            #Mutation (generate donor vector)
                mutant = p_{r1,G} + μ_{F1}·(p_{r2,G} − p_{r3,G}) + μ_{F2}·(p_{r4,G} − p_{r5,G})
            #Crossover (generate trail vector)
            Generate randomly, CR_{rand[0,1]} = randint[1; k]
            If      CR_{rand[0,1]} < μ_{CR}
                        Pop[i][position] = mutant
            Else
                        Pop[i][position] = Gbest[position]
            End If
            #End DE
            #Update particles velocity
              Pop[i][velocity] = Pop[i][velocity] + c_1 * rand
                        * (Pop[i][bestposition] − Pop[i][position]) + c_2 * rand
                        * (Gbest[position] − Pop[i][position])
            Apply Bounds
        Cost Evaluation (X^n_j, P^n_j) (%%RUN TRANUS)
            #Update best personal using
        If Pop[i][cost] < Pop[i][bestcost]
            Pop[i][bestposition] = Pop[i][position];
            Pop[i][bestcost] = Pop[i][cost];

            #Update best Global
            If    Pop[i][best] < Gbest[cost]
                  Gbest[position] = Pop[i][best]
                  Gbest[cost] = Pop[i][bestcost]
            End If
        End If
    End For
    Return bestcost[it] = Gbest[cost]
```

Like other stochastic algorithms, PSODE starts with initialization because it is PSO-based; all the processes are the same as PSO, except particle updating, whereby the DE operators "Mutation and Crossover" are merged into the process. During the mutation stage, random particles are perturbated, and results are used alongside Gbest for the crossover process. PSO operators take these new particles and continue the optimization process.

3.5.1.1 Parameter setting

All the five mutation techniques mentioned in section 3.2.4.1 are evaluated over the parameter ranges listed in Table 3-10 to get the best out of the DE algorithm. Through a semi-automatic evaluation process, using RMSE and MANE techniques, DE/rand/2 Equation 3.23 as the mutation strategy got the best results using the parameters listed in Table 3-11.

Table 3-10: Parameters settings range for the PSODE operators (RMSE and MANE)
Preglednica 3-10: Obseg nastavitev parametrov za operaterje PSODE (RMSE in MANE)

Parameters	Settings	Description
MaxIt	400 for MANE),1000 for RMSE	maximum number of generations
NP	[10, 100] In steps 5	initial population number
C_1	[0.1, 2.0] In steps 0.1	personal learning coefficient
C_2	[0.1, 2.0] In steps 0.1	global acceleration coefficient
μ_{F1}	[0.1, 2.0] In steps 0.1	mutation factor
μ_{F2}	[0.1, 2.0] In steps 0.1	mutation factor
Cr	[0.1, 2.0] In steps 0.1	crossover probability

Table 3-11: Parameters selected for PSODE operators (RMSE and MANE)
Preglednica 3-11: Parametri, izbrani za operaterje PSODE (RMSE in MANE)

Parameters (RMSE)	Parameters (MANE)	Description
MaxIt = 1000	*MaxIt* = 400	maximum number of generations
NP = 30	*NP* = 20	initial population number
C_1 = 1.5	C_1 =1.2	personal acceleration coefficient
C_2 = 1.3	C_2 = 1.2	global acceleration coefficient
w = 0.9	w = 0.9	inertia weight
μ_{F1} = 0.3	μ_{F1} = 0.3	mutation factor
μ_{F2} =0.6	μ_{F2} =0.6	mutation factor
Cr = 0.8	*Cr* = 0.9	crossover probability

3.5.2 Hybrid GADE algorithm

As mentioned, GA and DE algorithms follow the same operators with minor differences. Since GA employs steady-state replacement, it executes more quickly than other techniques. Every generation's worst chromosome gets swapped out for a better one. Regarding reaching the lowest value, DE occasionally outperforms other methods, and its generation update has substantial results regarding the generic chromosome of the preceding generation. On the other hand, DE moves more slowly than GA.

Literature reviews show several more weaknesses and advantages of the GA and DE algorithms, and to overcome their weaknesses and get the benefits of each one advantage, the researcher proposes the GADE hybrid algorithm. The version of the GADE hybrid algorithm applied to the study example of this thesis is inspired by [164]. This application GADE algorithm improves the optimization process and, in the meantime, gives another evaluation of the proposed calibration approach using the DE algorithm.

GA algorithm follows its standard process and operation, with the only difference being that the crossover operation is replaced with the DE algorithm perturbation method, and the GA steps follow the rest of the process.

3.5.2.1 Parameter setting

The mutation strategies listed in section 3.2.4.1 are evaluated over the parameter ranges listed in Table 3-12 to get the best out of the DE algorithm during the hybridization. This process is facilitated with a semi-automatic evaluation approach using RMSE and MANE techniques. The evaluation results show the best outcome with the DE/rand/1 as the mutation strategy using the parameters listed in Table 3-13.

Table 3-12: Parameters settings range for the GADE operators (RMSE and MANE)
Preglednica 3-12: Razpon nastavitev parametrov za operaterje GADE (RMSE in MANE)

Parameters	Settings	Description
MaxIt	400 for MANE),1000 for RMSE	maximum number of generations
NP	[10, 100] In steps 5	initial population number
Cr	[0.1, 1.0] In steps 0.1	crossover probability
mu	[0.1, 1.0] In steps 0.1	mutation rate
μ_{F1}	[0.1, 2.0] In steps 0.1	mutation factor
μ_{F2}	[0.1, 2.0] In steps 0.1	mutation factor

Table 3-13 Parameters selected for GADE operators (RMSE and MANE)
Preglednica 3-13: Parametri, izbrani za operaterje GADE (RMSE in MANE)

Parameters (RMSE)	Parameters (MANE)	Description
MaxIt = 1000	*MaxIt* = 400	maximum number of generations
NP = 30	*NP* = 20	initial population number
Cr = 1	*Cr* = 1	crossover probability
mu = 0.18	*mu* = 0.18	mutation rate
μ_{F1} = 0.3	μ_{F1} = 0.3	mutation factor
μ_{F2} =0.6	μ_{F2} =0.6	mutation factor

Algorithm 3-5: Calibration of land use model parameters using modified GADE algorithm

Initialization

Define Cost Functions [MANE and RMSE]
Generate a random initial population $\{x_{i,0} \mid i = 1,2,\ldots,NP\}$
For i in range (*NP*)
#Evaluate Cost function,
$\boldsymbol{fitness = \{f(x_{i,G}), i = 1, 2, \ldots, Np\}}$ %%$\boldsymbol{RUN\ TRANUS}$
End For
$best_{index} = arg\ min\ (fitness)$
$bestsol = x_{best}$

bestcost = np.empty(***MaxIt***)

Main Loop

For i in range (***MaxIt***):
Crossover Operation
For *j* in range (*nCrossover*)
#Start DE
Select parents randomly
$p_{r1,G}, p_{r2,G}, p_{r3,G}, p_{r4,G}, p_{r5,G}, \in [1; Np, replace = False]$
#Mutation (generate donor vector)
Generate *offspring*
$$C1 = p_{r1,G} + \mu_{F1}.(p_{r2,G} - p_{r3,G})$$
$$C2 = p_{r1,G} + \mu_{F2}.(p_{r4,G} - p_{r5,G})$$
#End DE
#Evaluate cost functions
Cost Evaluation ($\boldsymbol{X^n}_j$, $\boldsymbol{P^n}_j$) (%%$\boldsymbol{RUN\ TRANUS}$)
Update $bestsol$
End For
#Mutation Operation
For *j* in range (*nMutation*)
Select parents randomly
Create *offspring* using parents
#Evaluate cost functions
Cost Evaluation ($\boldsymbol{X^n}_j$, $\boldsymbol{P^n}_j$) (%%$\boldsymbol{RUN\ TRANUS}$)
Update $bestsol$
#Merge, Sort, and Selection
Population merging
Population sorting
Exclude extra population
Generate new population
#Update the best cost
bestcost[*it*] = *bestsol*
End For

3.6 Proposed Calibration Approach Flowchart and GUI

In the proposed methodology, as seen in the optimization process Flow-Chart of Figure 3-5, the total number of iterations (*MaxIter*) and the total number of populations (*NP*) are equally shared by DE, GA, PSO, PSODE and GADE algorithms to have a proper evaluation. To implement the iteration steps of the proposed flow chart, all mentioned algorithms are coded in Python, and the TRANUS land-use model (Example C, given by the TRANUS tutorial) is used as a case study to test the proposed calibration approach. After the implementation of the sensitivity, the proposed optimization techniques are performed to calibrate the desired economic parameters (Output of sensitivity analysis) alongside the shadow prices (price-correcting additive variables). Root Mean Square Error (RMSE) and Mean Absolute Normalized Error (MANE) have been employed as standard statistical metrics to measure the goodness of the proposed models. In this work, two stopping criteria for the proposed optimization techniques are defined: (1) if RMSE or MANE values were repeated more than n-times (Convergence check value = 40) or (2) if the difference between the current MANE or RMSE value and the average FITNESS value was less than 0.000001 (techniques precision value).

A graphical user interface (GUI) is created to prevent confusion over the code's content and desired calibration techniques and inputs (see Figure 3-6). The offered GUI makes it easier for the user to complete the calibration procedure and allows users to pick their chosen models, optimization measures, and optimization algorithms easily.

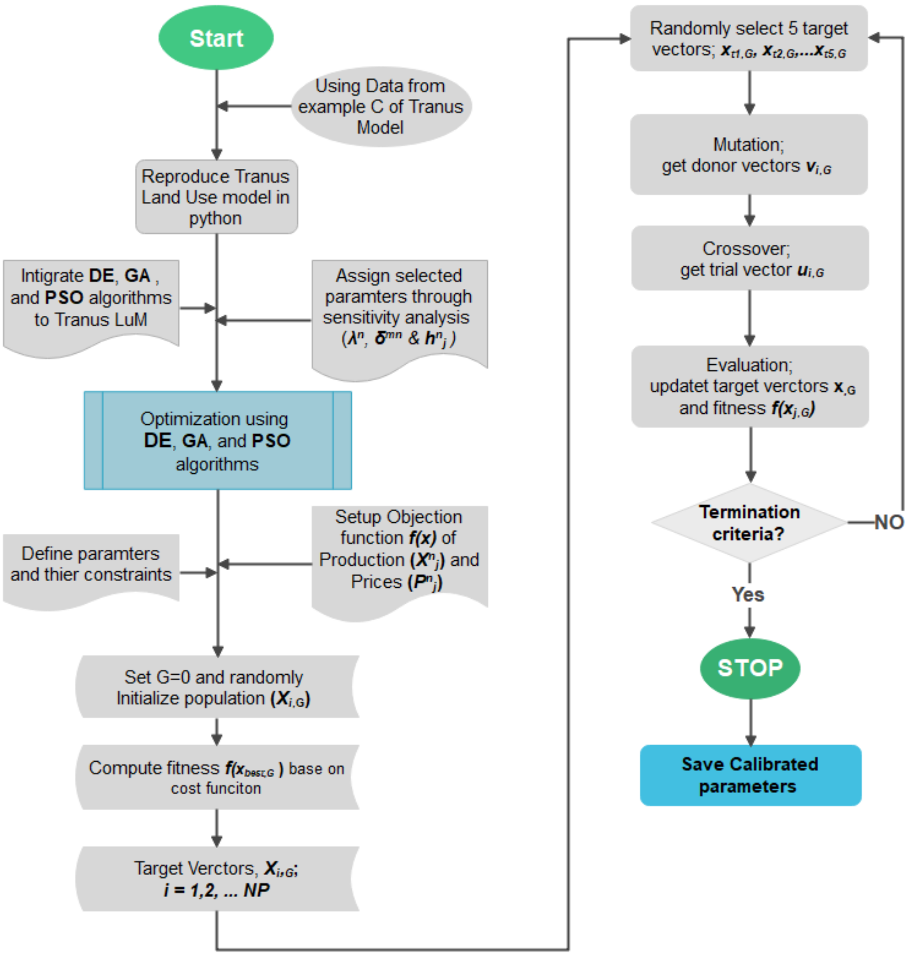

Figure 3-5: Calibration Approach Flow-Chart
Slika 3-5: Diagram poteka kalibracijskega pristopa

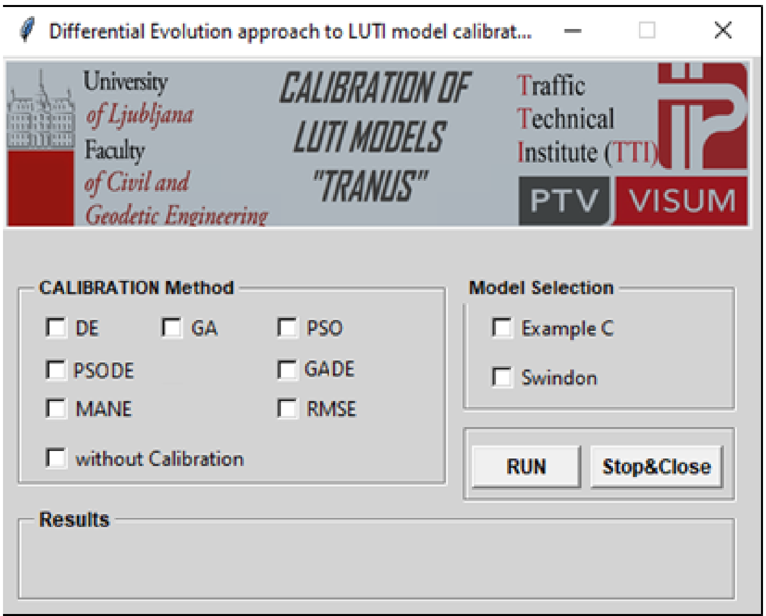

Figure 3-6: Proposed Calibration Approach GUI developed using Python
Slika 3-6: Predlagani GUI za pristop kalibracije, razvit z uporabo Pythona

4 RESULTS AND DISCUSSION

Let us consider an area with N sectors and M zones. Productions and prices are provided as observable data for a specific base year. The set of observed productions and price data were denoted by $\boldsymbol{X}^{act} \in \boldsymbol{R}^{NxM}$ and $\boldsymbol{P}^{act} \in \boldsymbol{R}^{NxM}$ Respectively. The proposed calibration approach developed in this thesis is tested using data from example C of the TRANUS tutorial. The region defined in this example was divided into three geographical zones $(j = 1, 2, 3)$ and five economic sectors $(n, m = 1, 2, 3, 4, 5)$. The economic sectors include essential employment, service employment, low-income households, high-income households, and land. Table 4-1 and Table 4-2 present the TRANUS default values of the selected parameters against their optimized values using the DE, GA, PSO, PSODE, and GADE algorithms as the optimization techniques and RMSE and MANE as multi-objective functions. Selected parameters are bounded based on the highest and lowest TRANUS default values as follows:
$\boldsymbol{\delta^{nm}}$ = (0.000001, 0.00001), $\boldsymbol{\lambda^{n}}$ (0, 1), and $\boldsymbol{h^{n}}_{j}$ = (-100, 0).

Table 4-1: Optimized values using the DE, GA, and PSO algorithms against TRANUS defaults.
Preglednica 4-1: Optimizirane vrednosti z uporabo DE, GA in PSO glede na privzete vrednosti TRANUS.

Parameter	TRANUS	DE_{RMSE}	DE_{MANE}	GA_{RMSE}	GA_{MANE}	PSO_{RMSE}	PSO_{MANE}
h^1_1	0.00	-15.29	-53.40	-32.09	-44.06	-40.47	-18.46
h^2_1	-50.11	-78.08	-66.34	-92.52	-62.62	-35.60	-46.70
h^3_1	-34.64	-28.85	-17.49	-1.03	-41.44	0.00	-35.56
h^4_1	-27.74	-15.03	-11.98	-58.65	0.00	-33.40	0.00
h^5_1	-8.10	-7.33	-16.73	-100.00	-29.06	0.00	-0.58
h^1_2	0.00	-37.53	-66.47	-0.05	-0.06	-15.44	-100.00
h^2_2	-36.92	-46.13	-35.40	-55.88	-49.01	-18.04	-12.19
h^3_2	-61.00	-52.42	-53.52	-9.73	-72.33	-43.02	-72.29
h^4_2	-65.69	-83.32	-58.24	-93.44	-35.92	-68.64	-69.88
h^5_2	-23.18	-10.04	-26.81	-100.00	-42.34	-22.32	-44.06
h^1_3	0.00	-9.62	-12.44	-30.61	-48.42	-3.10	-22.14
h^2_3	-34.40	-59.94	-32.94	-17.81	-46.11	-7.45	-10.96
h^3_3	-60.04	-49.63	-51.53	-11.33	-66.47	-37.25	-67.93
h^4_3	-52.84	-65.68	-42.34	-91.79	-21.17	-47.41	-50.74
h^5_3	-22.27	-21.71	-32.13	-42.14	-32.59	-0.07	-43.55
δ^{11}	0.0	0.0	0.0	0.0	0.0	0.0	0.0
δ^{21}	0.0	0.00000754	0.00000454	0.00000126	0.00000953	0.00000332	0.00000133
δ^{31}	0.0	0.00000217	0.00000344	0.00000872	0.00001	0.00000130	0.00000551
δ^{41}	0.0	0.00000955	0.00000383	0.000001	0.000001	0.00000963	0.000001
δ^{51}	0.000007	0.00000858	0.00000873	0.000001	0.00001	0.00000532	0.0000039
δ^{12}	0.0	0.00000591	0.00000779	0.000001	0.00001	0.00000943	0.000008
δ^{22}	0.0	0.0	0.0	0.0	0.0	0.0	0.0
δ^{32}	0.0	0.00000666	0.00000908	0.000001	0.00001	0.00000908	0.00000912
δ^{42}	0.0	0.00000285	0.00000881	0.00001	0.00000954	0.000001	0.00000763
δ^{52}	0.000008	0.00000720	0.00000876	0.00001	0.00001	0.00000910	0.00000998
δ^{13}	0.0	0.00000444	0.00000553	0.000001	0.00001	0.00000851	0.00001
δ^{23}	0.0	0.00000767	0.00000347	0.00000825	0.00001	0.000001	0.00000863
δ^{33}	0.0	0.0	0.0	0.0	0.0	0.0	0.0
δ^{43}	0.0	0.00000304	0.00000139	0.000001	0.00001	0.00000222	0.000001
δ^{53}	0.000007	0.00000542	0.00000731	0.00001	0.00001	0.00001	0.00001
δ^{14}	0.0	0.00000497	0.00000929	0.00000224	0.00001	0.00001	0.00001
δ^{24}	0.0	0.00000265	0.00000539	0.00000296	0.00001	0.00000554	0.00001
δ^{34}	0.0	0.00000657	0.00000826	0.00001	0.00001	0.00000466	0.000001
δ^{44}	0.0	0.0	0.0	0.0	0.0	0.0	0.0
δ^{54}	0.000006	0.00000679	0.00000754	0.00000984	0.00000632	0.00000333	0.00000879
δ^{15}	0.0	0.00000723	0.00000757	0.00000621	0.000010	0.000001	0.000001

δ^{25}	0.0	0.00000175	0.00000627	0.00000288	0.00000848	0.000001	0.00001
δ^{35}	0.0	0.00000306	0.00000591	0.000001	0.000010	0.000001	0.000001
δ^{45}	0.0	0.00000341	0.00000618	0.000001	0.00000888	0.00000780	0.00000846
δ^{55}	0.0	0.0	0.0	0.0	0.0	0.0	0.0
λ^1	1	0.4024	0.4172	1.0	0.0	0.16	0.96
λ^2	1	0.3328	0.793	0.750	1.0	0.33	0.35
λ^3	1	0.9735	0.7577	0.125	0.94	1.0	0.85
λ^4	1	0.5220	0.7063	0.588	1.0	0.77	0.60
λ^5	1	0.9752	0.8227	0.760	1.0	1.0	1.0

Table 4-2: Optimized values using the DE, PSODE, and GADE algorithms against TRANUS defaults
Preglednica 4-2: Optimizirane vrednosti z uporabo algoritmov DE, PSODE in GADE glede na privzete vrednosti TRANUS

Parameter	TRANUS	DE_{RMSE}	DE_{MANE}	$PSODE_{RMSE}$	$PSODE_{MANE}$	$GADE_{RMSE}$	$GADE_{MANE}$
h^1_1	0.00	-15.29	-53.40	-79.79	-93.96	-57.74	-45.35
h^2_1	-50.11	-78.08	-66.34	-35.70	-25.79	-58.93	-72.80
h^3_1	-34.64	-28.85	-17.49	-0.52	-0.39	-25.71	-42.58
h^4_1	-27.74	-15.03	-11.98	-1.23	-1.06	-60.01	-39.49
h^5_1	-8.10	-7.33	-16.73	-100.00	-8.89	-100.00	-20.73
h^1_2	0.00	-37.53	-66.47	-73.28	0.00	-99.44	-57.35
h^2_2	-36.92	-46.13	-35.40	-3.87	-4.21	-42.25	-19.54
h^3_2	-61.00	-52.42	-53.52	-13.90	-22.43	-56.14	-78.86
h^4_2	-65.69	-83.32	-58.24	-14.15	-39.52	-83.16	-93.89
h^5_2	-23.18	-10.04	-26.81	-100.00	-1.86	-100.00	-44.44
h^1_3	0.00	-9.62	-12.44	-98.79	-58.38	-88.34	-56.53
h^2_3	-34.40	-59.94	-32.94	-2.19	-1.22	-44.62	-19.88
h^3_3	-60.04	-49.63	-51.53	-21.39	-25.36	-54.56	-72.78
h^4_3	-52.84	-65.68	-42.34	-8.67	-23.99	-75.23	-83.49
h^5_3	-22.27	-21.71	-32.13	-100.00	-0.04	-100.00	-44.05
δ^{11}	0.0	0.0	0.0	0.0	0.0	0.0	0.0
δ^{21}	0.0	0.00000754	0.00000454	0.00000580	0.00000861	0.00000604	0.00000190
δ^{31}	0.0	0.00000217	0.00000344	0.00000949	0.00000145	0.00001000	0.00000917
δ^{41}	0.0	0.00000955	0.00000383	0.00000989	0.00000954	0.00000100	0.00000190
δ^{51}	0.000007	0.00000858	0.00000873	0.00000482	0.00001	0.00000559	0.00000999
δ^{12}	0.0	0.00000591	0.00000779	0.00000104	0.00000966	0.00001000	0.00000899
δ^{22}	0.0	0.0	0.0	0.0	0.0	0.0	0.0
δ^{32}	0.0	0.00000666	0.00000908	0.00000760	0.00000261	0.00000460	0.00000910
δ^{42}	0.0	0.00000285	0.00000881	0.00000708	0.00000100	0.00000640	0.00000251
δ^{52}	0.000008	0.00000720	0.00000876	0.000001	0.00000927	0.00000476	0.00000549
δ^{13}	0.0	0.00000444	0.00000553	0.00000180	0.00000107	0.00000619	0.00000100
δ^{23}	0.0	0.00000767	0.00000347	0.00000192	0.00000134	0.00000100	0.00000651
δ^{33}	0.0	0.0	0.0	0.0	0.0	0.0	0.0
δ^{43}	0.0	0.00000304	0.00000139	0.00000285	0.00000532	0.00001000	0.00000977
δ^{53}	0.000007	0.00000542	0.00000731	0.00000942	0.00000538	0.00000471	0.00000999
δ^{14}	0.0	0.00000497	0.00000929	0.00000993	0.00000100	0.00000125	0.00000417
δ^{24}	0.0	0.00000265	0.00000539	0.00000551	0.00000554	0.00000100	0.00000190
δ^{34}	0.0	0.00000657	0.00000826	0.00000580	0.00000422	0.00000100	0.00000217
δ^{44}	0.0	0.0	0.0	0.0	0.0	0.0	0.0
δ^{54}	0.000006	0.00000679	0.00000754	0.00000698	0.00000457	0.00001000	0.00000871
δ^{15}	0.0	0.00000723	0.00000757	0.00000198	0.00000235	0.00000976	0.00000206
δ^{25}	0.0	0.00000175	0.00000627	0.00000804	0.00000249	0.00000871	0.00000100
δ^{35}	0.0	0.00000306	0.00000591	0.00000222	0.00000951	0.00000290	0.00000759
δ^{45}	0.0	0.00000341	0.00000618	0.00000914	0.00000999	0.00000936	0.00000918
δ^{55}	0.0	0.0	0.0	0.0	0.0	0.0	0.0
λ^1	1	0.4024	0.4172	0.26	0.04	0.83	0.83
λ^2	1	0.3328	0.793	0.28	1.00	0.99	0.18
λ^3	1	0.9735	0.7577	0.58	0.91	0.47	0.79
λ^4	1	0.5220	0.7063	0.92	0.89	0.98	0.66
λ^5	1	0.9752	0.8227	0.45	0.97	0.92	0.99

The calibrated values of the parameters presented in Table 4-1 and Table 4-2 show that using GA and PSO detect lower or upper bounds of the parameters as the optimum results, such as GA_{RMSE} ($h^5{}_1 = h^5{}_2 = -100$), GA_{MANE} ($h^4{}_1 = 0.00$, $\delta^3{}_1 = 0.00001$, $\delta^4{}_1 = 0.000001$, $\delta^5{}_1 = 0.0001$) and PSO_{MANE} ($h^4{}_1 = 0.00$, $h^1{}_2 = -100.00$, $\delta^4{}_3 = 0.000001$, $\delta^5{}_3 = 0.00001$, $\delta^1{}_4 = 0.00001$), PSO_{RMSE} ($h^3{}_1 = 0.00$, $\delta^{42} = 0.000001$, $\delta^{53} = \delta^{14} = 0.00001$, $\lambda^3 = \lambda^5 = 1.0$) so they were not able to improve MANE and RMSE values further. While these sticking values are not seen using the DE algorithm with MANE and RMSE, the model continuously improves the parameter value. A further improvement on GA and PSO algorithms, using hybrid GADE and PSODE methods, is also helping to improve the calibration approach as shown in Table 4-2, where the number of sucking values is minimum than GA and PSO presented in Table 4-1. The effect of these sticking to upper and lower range boundaries can also be visible with the modeled values of the Production and Prices that are presented in the following tables, where the modeled and observed data are compared.

The evolution of MANE and RMSE values using DE, GA, PSO, PSODE, and GADE optimizations are depicted in Figure 4-1 and Figure 4-2, in which their best-cost values are shown using colored lines according to the legend details.

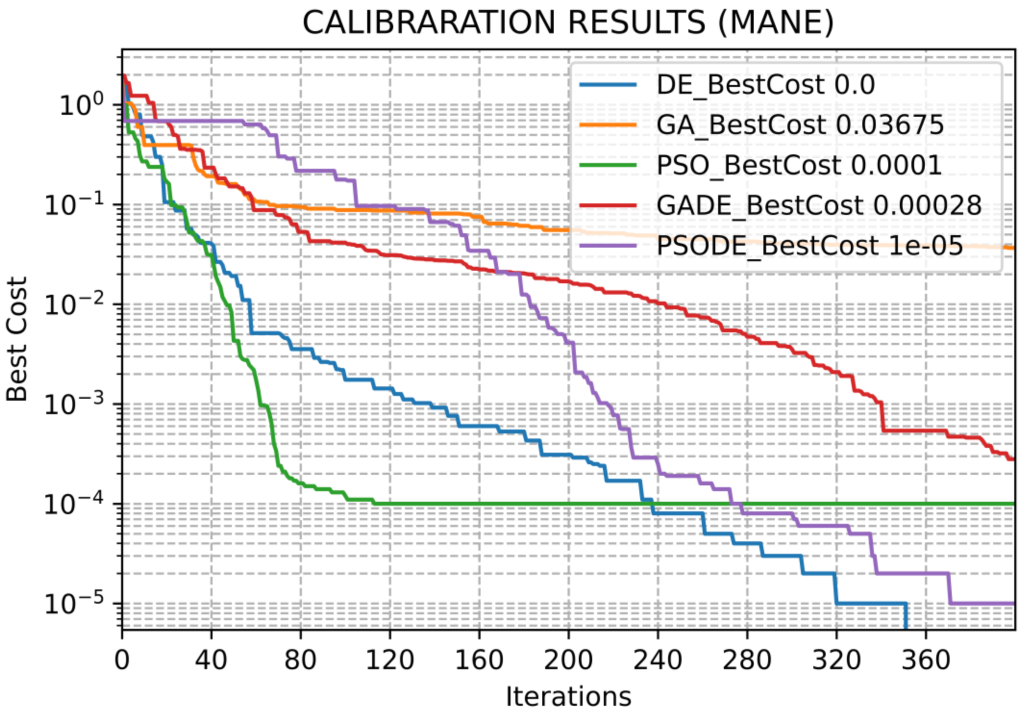

Figure 4-1: MANE values of DE, GA, PSO, PSODE, and GADE optimizations
Slika 4-1: Vrednosti MANE optimizacij DE, GA, PSO, PSODE in GADE

Figure 4-1 clearly shows that the DE algorithm outperformed the GA and PSO algorithms and the Hybrid of GADE and PSODE algorithms. Using the DE algorithm as an optimization method, the MANE value almost reached zero, while MANE values obtained using GA, PSO GADE, and PSODE were 0.00044, 0.0001, 0.00028, and 0.00005, respectively. Using hybrid algorithms shows a significant improvement in GA and PSO, while the DE algorithm still has an outstanding performance.

All algorithms started with a similar MANE value. Although PSO reached its optimum value sooner (after only 110 iterations), it could not improve it more, while all other algorithms proceeded to improve MANE values. DE and GA algorithms reached their optimum MANE values after 350 and 400 iterations. GADE is after 400 iterations, and PSODE is after 380 iterations.

Based on the MANE multi-objective function (price and production) values, the DE algorithm produced the best results and outperformed all the other algorithms. However, in terms of speed, PSO algorithms have the fastest convergence and outperform their competitors.

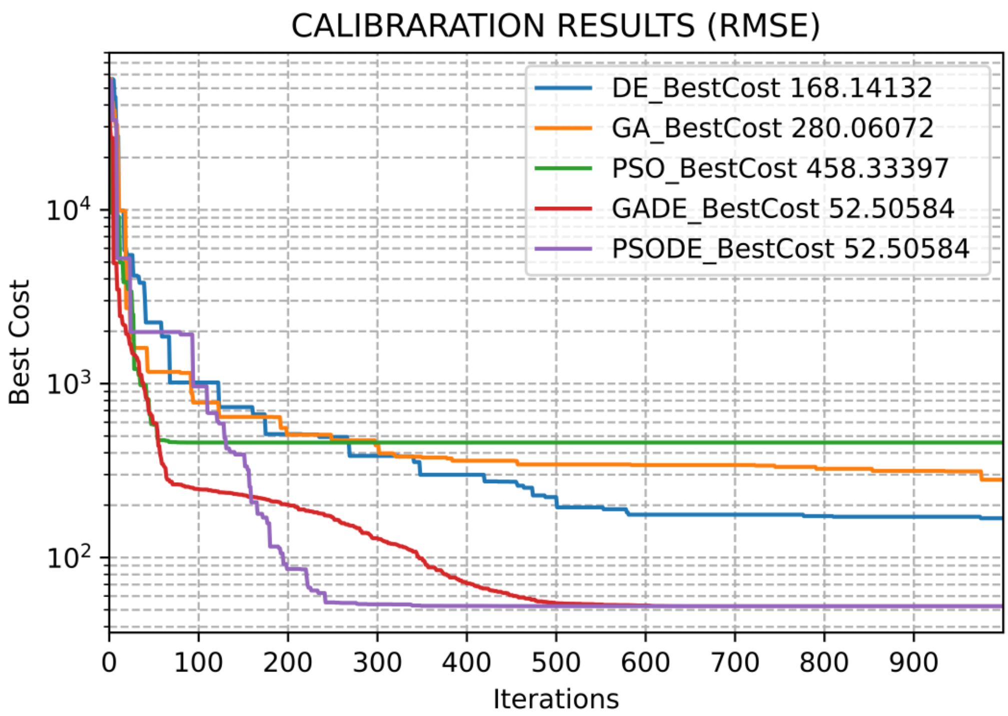

Figure 4-2: RMSE values of DE, GA, PSO, PSODE, and GADE optimizations
Slika 4-2: RMSE vrednosti optimizacij DE, GA, PSO, PSODE in GADE

Figure 4-2 shows that the best RMSE value was obtained by the DE algorithm with a value of 168. However, neither GA nor PSO could conclude that the RMSE values were less than 280 and 458, respectively. Using hybrid algorithms, both GADE and PSODE significantly improve the RMSE cost values, with the same value of 52.5084. Here again, all three optimization techniques started with a similar RMSE value. However, PSO stopped improving before 100 iterations, while GA and DE continuously improved RMSE values and ended almost 1000 iterations.

The outstanding performances of the hybrid algorithms do not guarantee that they also have significant parameters' modeled values. The comparisons of the modeled values against the observed ones prove that, despite the outstanding performance of the hybrid algorithms (GADE and PSODE), reaching the minimum RMSE values, the Production and Prices modeled values are not close enough to the Observed ones, as it is obtained using DE algorithm. The consistency of the calibration algorithm provided here is demonstrated. Because the DE estimator employing both MANE and RMSE is asymptotically efficient, the estimate becomes closer to the actual solution as the number of iterations increases, as seen in Figure 4-1 and Figure 4-2. Furthermore, the optimal cost values were continually reduced by the DE technique. In addition, DE discovered better solutions almost in every iteration. As a result, we can state that the proposed calibration technique can reliably estimate the land-use characteristics. The calibration results of multi-objective functions (prices and productions) for DE, GA, PSO, hybrid GADE, and PSODE optimization techniques are presented in Table 4-3, Table 4-4, Table 4-5, Table 4-6, and Table 4-7, respectively. The TRANUS results are referred to as land-use (observed) data proposed by TRANUS, and MANE-based and RMSE-based values are referred to as values obtained by the calibration model using DE, GA, and PSO algorithms. Mod. /Obs. The ratio is also referred to as modeled values obtained by MANE and RMSE compared to the observed value given by TRANUS. Mod. /Ob. the ratio of productions (***X"j***) and prices (***P"j***) using the DE algorithm as calibration technique while MANE and RMSE as objective functions are presented in Figure 4-3.

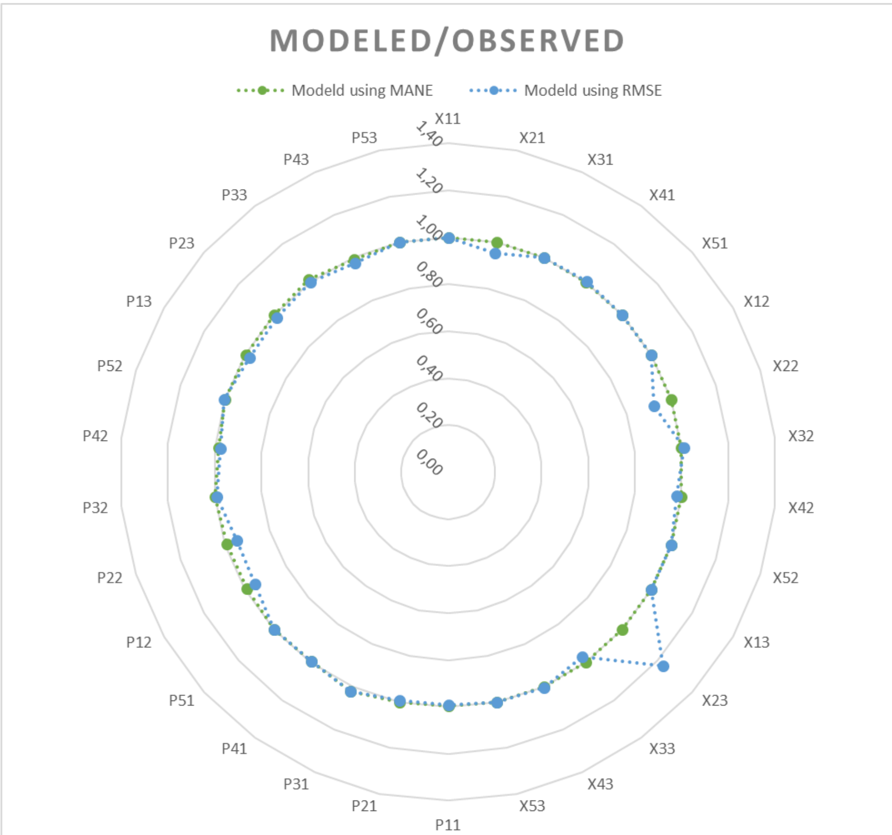

Figure 4-3: Mod./Ob. the ratio of Production and Prices using DE
Slika 4-3: Mod./Ob. razmerje med proizvodnjo in cenami z uporabo DE

Table 4-3: Observed prices and productions vs. model values using the DE algorithm.
Preglednica 4-3: Opazovane cene in produkcije v primerjavi z vrednostmi modelov z uporabo algoritma DE

Parameter	Observed	Model using MANE	Mod. /Ob.	Model using RMSE	Mod. /Ob.
X^1_1	5.000	5.000,00	1,00	5.000,00	1,00
X^2_1	3.500	3.500,00	1,00	3.345,14	0,96
X^3_1	4.000	4.000,00	1,00	4.003,45	1,00
X^4_1	1.500	1.500,00	1,00	1.505,91	1,00
X^5_1	066	66,00	1,00	66,00	1,00
X^1_2	800	800,00	1,00	800,00	1,00
X^2_2	700	700,00	1,00	644,29	0,92
X^3_2	13.000	13.000,00	1,00	13.124,86	1,01
X^4_2	3.000	3.000,00	1,00	2.936,47	0,98
X^5_2	110	110,00	1,00	110,00	1,00
X^1_3	1.100	1.100,00	1,00	1.100,00	1,00
X^2_3	900	900,00	1,00	1.110,58	1,23
X^3_3	5.000	5.000,00	1,00	4.871,68	0,97
X^4_3	11.500	11.500,00	1,00	11.557,62	1,01
X^5_3	128	128,00	1,00	128,00	1,00
P^1_1	14.546	14.506,38	1,00	14.370,79	0,99
P^2_1	14.191	14.195,38	1,00	14.122,73	0,99
P^3_1	2.705	2.769,40	1,02	2.835,52	1,02
P^4_1	3.862	3.840,60	0,99	3.815,62	0,99
P^5_1	250.000	250.000,18	1,00	249.942,40	1,00
P^1_2	11.973	11.875,61	0,99	11.299,58	0,95
P^2_2	11.714	11.642,09	0,99	11.018,42	0,95
P^3_2	2.447	2.446,64	1,00	2.429,64	0,99
P^4_2	3.341	3.289,97	0,98	3.215,25	0,98
P^5_2	120.000	120.000,05	1,00	120.364,69	1,00
P^1_3	12.197	12.166,74	1,00	11.916,16	0,98
P^2_3	11.703	11.710,39	1,00	11.493,34	0,98
P^3_3	2.644	2.677,51	1,01	2.675,75	1,00
P^4_3	3.693	3.656,64	0,99	3.571,54	0,98
P^5_3	180.000	180.000,03	1,00	180.126,83	1,00

The outstanding performance of the DE algorithm using MANE as the objective function is proved by the results presented in Table 4-3 and Figure 4-3. Using the MANE objective function, the DE calibration technique enables us to reach the observed production values without any error and observed price values with a slight difference, whereas the variation of the results Mod. /Obs. ratios came to between P^4_2 **= 0.98** and P^4_1 **=1.02**. However, the DE calibration technique using the RMSE objective function can model the observed price and production values with a slightly higher discrepancy, with results variation ratios between $X^2{}_2$ = **0.92** and $P^3{}_1$ = **1.02.** Only the parameter ($X^2{}_3$) was not modeled properly as its Mod. /Obs. ratio is 1.23.

As seen in Table 4-4 and Figure 4-4, the GA calibration technique using the MANE objective function precisely modeled the observed production values. In contrast, price values slightly differed with a variation ratio between $X^3{}_1$ =***0.98*** and $P^5{}_1$ = **1.05**. However, the GA calibration technique using the RMSE objective function had a significant discrepancy, where the variations of the results were between $X^4{}_1$ = **0.78** and $X^5{}_1$ = **1.91**. These results indicate the deficiency of the GA calibration technique using RMSE objective functions.

Table 4-4: Observed prices and productions vs. model values using the GA algorithm.
Preglednica 4-4: Opazovane cene in produkcije v primerjavi z vrednostmi modelov z uporabo algoritma GA

Parameter	Observed	Model using MANE	Mod. /Ob.	Model using RMSE	Mod. /Ob.
$X^1{}_1$	5.000	5.000,00	**1,00**	5.000,00	**1,00**
$X^2{}_1$	3.500	3.498,39	**1,00**	3.344,03	**0,96**
$X^3{}_1$	4.000	3.902,24	**0,98**	3.925,01	**1,01**
$X^4{}_1$	1.500	1.580,20	**1,05**	1.232,91	**0,78**
$X^5{}_1$	066	66,00	**1,00**	126,29	**1,91**
$X^1{}_2$	800	800,00	**1,00**	800,00	**1,00**
$X^2{}_2$	700	700,24	**1,00**	948,97	**1,36**
$X^3{}_2$	13.000	13.121,71	**1,01**	12.881,36	**0,98**
$X^4{}_2$	3.000	2.987,37	**1,00**	2.908,93	**0,97**
$X^5{}_2$	110	110,00	**1,00**	154,50	**1,40**
$X^1{}_3$	1.100	1.100,00	**1,00**	1.100,00	**1,00**
$X^2{}_3$	900	901,37	**1,00**	806,99	**0,90**
$X^3{}_3$	5.000	4.976,05	**1,00**	5.193,63	**1,04**
$X^4{}_3$	11.500	11.432,44	**0,99**	11.858,16	**1,04**
$X^5{}_3$	128	128,00	**1,00**	128,00	**1,00**
$P^1{}_1$	14.546	14.670,09	**1,01**	17.684,01	**1,21**
$P^2{}_1$	14.191	14.356,46	**1,01**	17.241,20	**1,20**
$P^3{}_1$	2.705	2.762,40	**1,02**	4.113,98	**1,49**
$P^4{}_1$	3.862	3.920,03	**1,02**	5.692,39	**1,45**
$P^5{}_1$	250.000	261.486,92	**1,05**	250.000,00	**0,96**
$P^1{}_2$	11.973	12.091,49	**1,01**	13.801,88	**1,14**
$P^2{}_2$	11.714	11.921,34	**1,02**	13.383,06	**1,12**
$P^3{}_2$	2.447	2.469,73	**1,01**	2.966,45	**1,20**
$P^4{}_2$	3.341	3.357,30	**1,01**	3.995,44	**1,19**
$P^5{}_2$	120.000	120.998,30	**1,01**	120.000,00	**0,99**
$P^1{}_3$	12.197	12.306,26	**1,01**	14.000,72	**1,14**
$P^2{}_3$	11.703	11.877,96	**1,01**	12.778,78	**1,08**
$P^3{}_3$	2.644	2.674,38	**1,01**	2.895,27	**1,08**
$P^4{}_3$	3.693	3.710,30	**1,00**	3.948,53	**1,06**
$P^5{}_3$	180.000	179.726,42	**1,00**	181.233,79	**1,01**

Mod. /Ob. the ratio of productions (***X^n j***) and prices (***P^n j***) using hybrid GA algorithm as calibration technique while MANE and RMSE as objective functions are presented in Figure 4-4.

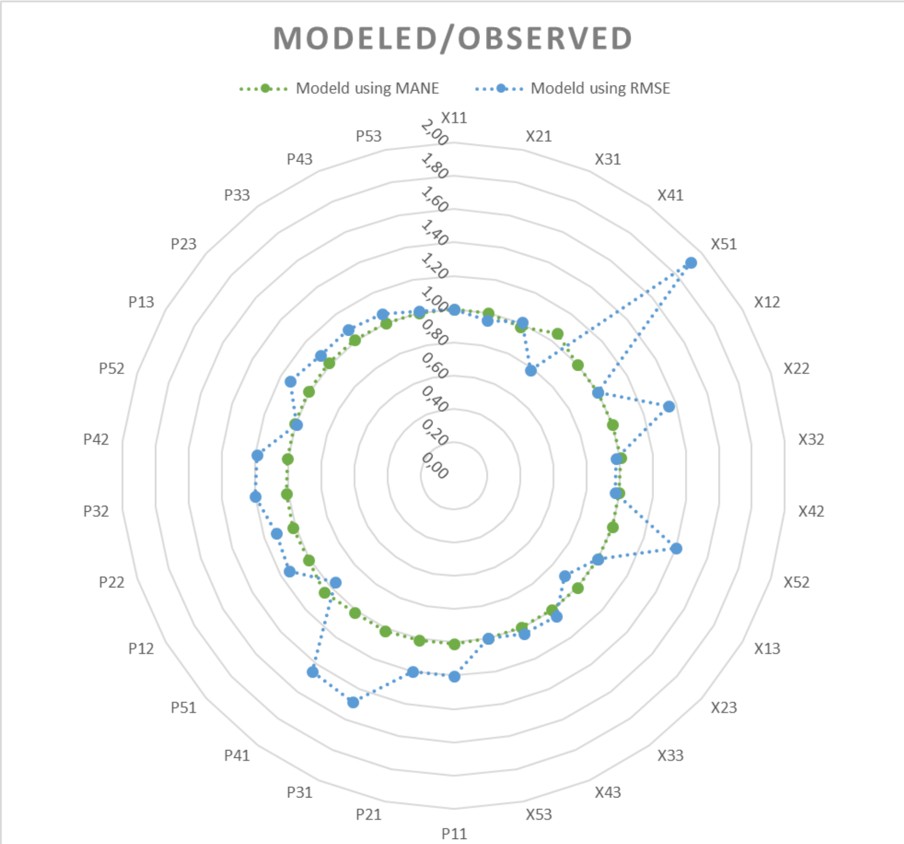

Figure 4-4: Mod./Ob. the ratio of Production and Prices using GA
Slika 4-4: Mod./Ob. razmerje med proizvodnjo in cenami z uporabo GA

Table 4-5: Observed prices and productions vs. model values using the PSO algorithm.
Preglednica 4-5: Opazovane cene in produkcije v primerjavi z vrednostmi modelov z uporabo algoritma PSO

Parameter	Observed	Model using MANE	Mod. /Ob.	Model using RMSE	Mod. /Ob.
X^1_1	5.000	5.000,00	**1,00**	5.000,00	**1,00**
X^2_1	3.500	3.499,99	**1,00**	3.308,86	**0,95**
X^3_1	4.000	4.000,76	**1,00**	3.513,84	**0,88**
X^4_1	1.500	1.500,00	**1,00**	1.754,36	**1,17**
X^5_1	066	66,00	**1,00**	66,00	**1,00**
X^1_2	800	800,00	**1,00**	800,00	**1,00**
X^2_2	700	700,00	**1,00**	829,76	**1,19**
X^3_2	13.000	12.999,81	**1,00**	13.531,57	**1,04**
X^4_2	3.000	2.999,98	**1,00**	3.160,15	**1,05**
X^5_2	110	110,00	**1,00**	110,00	**1,00**
X^1_3	1.100	1.100,00	**1,00**	1.100,00	**1,00**
X^2_3	900	900,01	**1,00**	961,38	**1,07**
X^3_3	5.000	4.999,43	**1,00**	4.954,58	**0,99**
X^4_3	11.500	11.500,03	**1,00**	11.085,50	**0,96**

X^5_3	128	128,00	**1,00**	128,00	**1,00**
P^1_1	14.546	14.459,62	**0,99**	14.431,80	**0,99**
P^2_1	14.191	13.734,56	**0,97**	13.971,27	**0,98**
P^3_1	2.705	2.517,07	**0,93**	2.533,63	**0,94**
P^4_1	3.862	3.564,76	**0,92**	4.153,04	**1,08**
P^5_1	250.000	250.001,69	**1,00**	250.000,00	**1,00**
P^1_2	11.973	11.564,76	**0,97**	11.690,30	**0,98**
P^2_2	11.714	11.138,65	**0,95**	11.378,38	**0,97**
P^3_2	2.447	2.378,41	**0,97**	2.315,10	**0,95**
P^4_2	3.341	3.245,70	**0,97**	3.405,31	**1,02**
P^5_2	120.000	120.001,82	**1,00**	120.000,00	**1,00**
P^1_3	12.197	12.134,86	**0,99**	11.993,28	**0,98**
P^2_3	11.703	11.363,51	**0,97**	11.433,90	**0,98**
P^3_3	2.644	2.570,76	**0,97**	2.405,07	**0,91**
P^4_3	3.693	3.587,16	**0,97**	3.778,45	**1,02**
P^5_3	180.000	180.003,26	**1,00**	180.000,00	**1,00**

Mod. /Ob. ratio of productions ($X^n j$) and prices ($P^n j$) using hybrid PSO algorithm as calibration technique while MANE and RMSE as objective functions are presented in Figure 4-5.

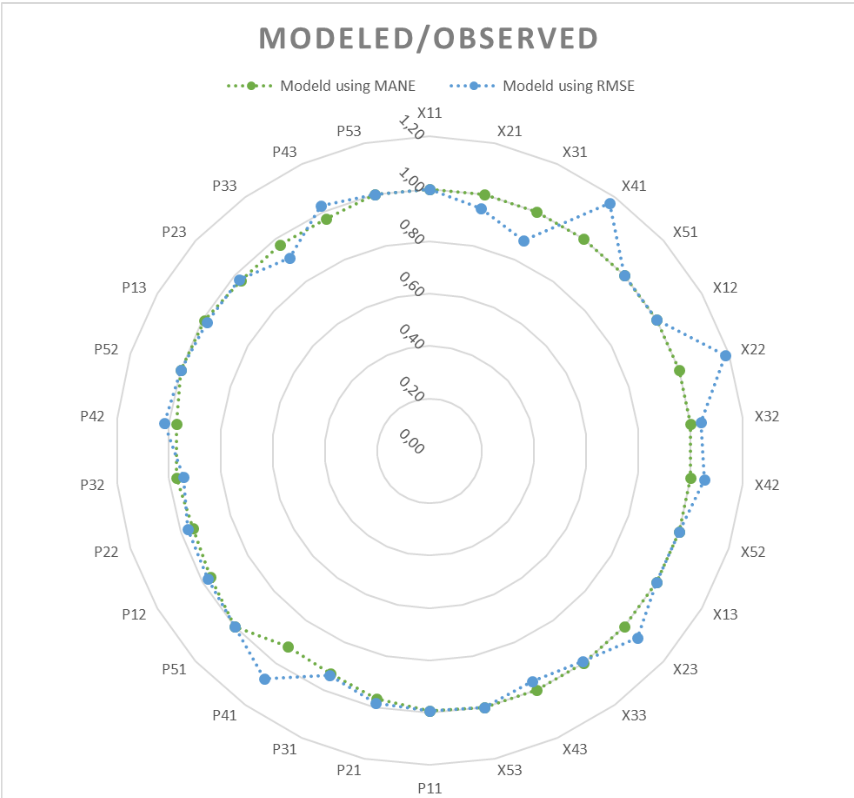

Figure 4-5: Mod./Ob. ratio of Production and Prices using PSO
Slika 4-5: Mod./Ob. razmerje med proizvodnjo in cenami z uporabo PSO

Considering the results presented in Table 4-5 and Figure 4-5, like both DE and GA results, the PSO calibration technique using the MANE objective function outperformed the PSO calibration technique using the RMSE objective function to model the observed data and utilizing MANE as an objective function the Mod. /Obs. The ratio ranges between $\boldsymbol{P^4{}_1}$ **= 0.92** and **1.0**, while utilizing RMSE has a higher error, varied between $\boldsymbol{X^3{}_1}$ **= 0.88** and $\boldsymbol{X^2{}_2}$ **= 1.19.**

Table 4-6: Observed prices and productions vs model values using the PSODE algorithm
Preglednica 4-6: Opazovane cene in produkcije v primerjavi z vrednostmi modelov z uporabo algoritma PSODE

Parameter	Observed	Model using MANE	Mod. /Ob.	Model using RMSE	Mod. /Ob.
$X^1{}_1$	5.000	5.000,00	**1,00**	5.000,00	**1,00**
$X^2{}_1$	3.500	3.500,02	**1,00**	3.503,27	**1,00**
$X^3{}_1$	4.000	3.999,87	**1,00**	4.000,45	**1,00**
$X^4{}_1$	1.500	1.500,00	**1,00**	1.500,16	**1,00**
$X^5{}_1$	066	66,00	**1,00**	131,53	**1,99**
$X^1{}_2$	800	800,00	**1,00**	800,00	**1,00**
$X^2{}_2$	700	699,98	**1,00**	700,15	**1,00**
$X^3{}_2$	13.000	13.000,07	**1,00**	13.000,40	**1,00**
$X^4{}_2$	3.000	2.999,94	**1,00**	3.000,36	**1,00**
$X^5{}_2$	110	110,00	**1,00**	154,31	**1,40**
$X^1{}_3$	1.100	1.100,00	**1,00**	1.100,00	**1,00**
$X^2{}_3$	900	899,99	**1,00**	896,58	**1,00**
$X^3{}_3$	5.000	5.000,06	**1,00**	4.999,16	**1,00**
$X^4{}_3$	11.500	11.500,07	**1,00**	11.499,49	**1,00**
$X^5{}_3$	128	128,00	**1,00**	197,06	**1,54**
$P^1{}_1$	14.546	14.618,71	**1,01**	19.777,10	**1,36**
$P^2{}_1$	14.191	14.344,00	**1,01**	19.352,84	**1,36**
$P^3{}_1$	2.705	2.853,27	**1,05**	4.318,38	**1,60**
$P^4{}_1$	3.862	4.081,90	**1,06**	5.952,72	**1,54**
$P^5{}_1$	250.000	250.027,51	**1,00**	250.000,00	**1,00**
$P^1{}_2$	11.973	12.206,26	**1,02**	16.426,90	**1,37**
$P^2{}_2$	11.714	11.999,85	**1,02**	16.126,53	**1,38**
$P^3{}_2$	2.447	2.479,20	**1,01**	3.325,98	**1,36**
$P^4{}_2$	3.341	3.384,30	**1,01**	4.453,35	**1,33**
$P^5{}_2$	120.000	120.030,98	**1,00**	120.000,00	**1,00**
$P^1{}_3$	12.197	12.211,17	**1,00**	16.927,34	**1,39**
$P^2{}_3$	11.703	11.789,83	**1,01**	16.371,01	**1,40**
$P^3{}_3$	2.644	2.678,16	**1,01**	3.798,96	**1,44**
$P^4{}_3$	3.693	3.742,81	**1,01**	5.164,40	**1,40**
$P^5{}_3$	180.000	180.015,74	**1,00**	180.000,00	**1,00**

Mod. /Ob. ratio of productions (***X^n j***) and prices (***P^n j***) using hybrid PSODE algorithm as calibration technique while MANE and RMSE as objective functions are presented in Figure 4-6.

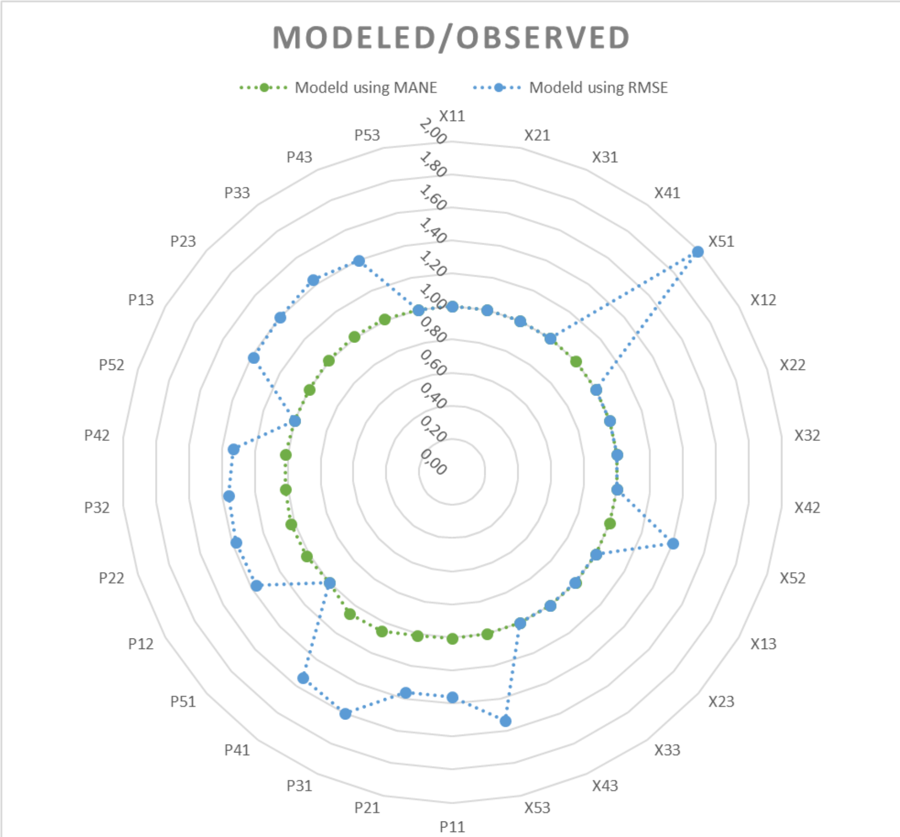

Figure 4-6: Mod./Ob. Ratio of Production and Prices using PSODE
Slika 4-6: Mod./Ob. razmerje med proizvodnjo in cenami z uporabo PSODE

The results presented in Table 4-6 and Figure 4-6 demonstrate the actual data of TRANUS and the modeled data of the PSODE algorithm using both MANE and RMSE for the Production and Prices. The parameter results show that the model has done a reasonably good job of predicting most of the actual data, as the parameter values using both MANE and RMSE are relatively close to the actual values. However, for some parameters like X^5_1 and X^5_3, P^1_1, and P^2_1, the RMSE values are much higher than the actual values, indicating that the model cannot predict these values accurately. Furthermore, the results suggest that most of the values are very close to the actual ones; using MANE as a multi-objective function demonstrates that the model has accurately predicted these values. In summary, while the model has reasonably predicted most of the actual values using MANE, it may require further calibration to improve its accuracy, especially for some parameters with higher RMSE values.

Table 4-7: Observed prices and productions vs model values using the GADE algorithm
Preglednica 4-7: Opazovane cene in produkcije v primerjavi z vrednostmi modelov z uporabo algoritma GADE

Parameter	Observed	Model using MANE	Mod. /Ob.	Model using RMSE	Mod. /Ob.
X^1_1	5.000	5.000,00	1,00	5.000,00	1,00
X^2_1	3.500	3.499,93	1,00	3.499,57	1,00
X^3_1	4.000	4.000,10	1,00	3.999,83	1,00
X^4_1	1.500	1.500,60	1,00	1.499,93	1,00
X^5_1	066	66,00	1,00	131,49	1,99
X^1_2	800	800,00	1,00	800,00	1,00
X^2_2	700	700,03	1,00	700,61	1,00
X^3_2	13.000	12.999,68	1,00	13.000,35	1,00
X^4_2	3.000	2.999,39	1,00	3.000,41	1,00
X^5_2	110	110,00	1,00	154,31	1,40
X^1_3	1.100	1.100,00	1,00	1.100,00	1,00
X^2_3	900	900,04	1,00	899,82	1,00
X^3_3	5.000	5.000,22	1,00	4.999,81	1,00
X^4_3	11.500	11.500,01	1,00	11.499,66	1,00
X^5_3	128	128,00	1,00	197,09	1,54
P^1_1	14.546	14.359,44	0,99	19.719,38	1,36
P^2_1	14.191	14.367,74	1,01	19.299,61	1,36
P^3_1	2.705	2.681,52	0,99	4.283,82	1,58
P^4_1	3.862	3.802,20	0,98	5.908,71	1,53
P^5_1	250.000	249.964,66	1,00	250.000,00	1,00
P^1_2	11.973	11.174,17	0,93	15.547,04	1,30
P^2_2	11.714	11.006,89	0,94	15.203,59	1,30
P^3_2	2.447	2.421,56	0,99	3.304,52	1,35
P^4_2	3.341	3.303,14	0,99	4.426,01	1,32
P^5_2	120.000	120.023,31	1,00	120.000,00	1,00
P^1_3	12.197	11.924,02	0,98	16.700,12	1,37
P^2_3	11.703	11.695,08	1,00	16.135,52	1,38
P^3_3	2.644	2.624,30	0,99	3.779,01	1,43
P^4_3	3.693	3.659,85	0,99	5.139,00	1,39
P^5	180.000	180.029,22	1,00	180.000,00	1,00

Figure 4-7 presents the ratio of modeled values of productions (***X*ⁿ*j***) and prices (***P*ⁿ*j***) using the hybrid GADE algorithm as a calibration method, while MANE and RMSE are objective functions.

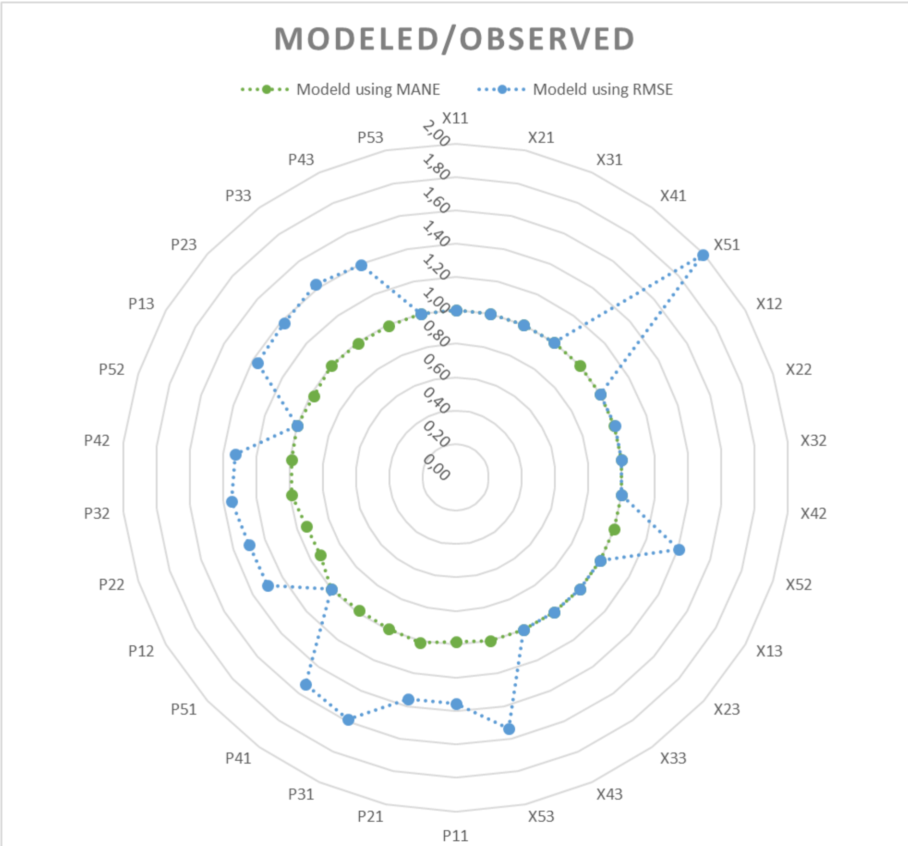

Figure 4-7: Mod./Ob. ratio of Production and Prices using GADE
Slika 4-7: Mod./Ob. razmerje med proizvodnjo in cenami z uporabo GADE

Based on the comparison of results presented in Table 4-7 and Figure 4-7, it is evident that utilizing MANE in the hybrid GADE model generally yields superior outcomes compared to using the RMSE. The MANE values for all parameters consistently demonstrate lower values than the corresponding RMSE values, indicating a better alignment between the GADE model and the TRANUS data.

GADE employing MANE exhibits a significant improvement in fitting the TRANUS results, suggesting its capability to capture overall trends and patterns within the data. Conversely, GADE utilizing RMSE tends to exhibit more significant errors, indicating a poorer fit to the actual data. This discrepancy may arise from RMSE's sensitivity to outliers and more significant deviations from the actual data, potentially leading to overfitting of the GADE model.

In summary, while GADE with MANE appears to provide a better fit to the TRANUS data compared to GADE with RMSE, it is crucial to exercise caution and comprehensively evaluate both models' performance before drawing any definitive conclusions.

The technique of creating a new population of solutions by perturbing solutions from the prior population is one of the critical distinctions between the three algorithms described above. The GA algorithm chooses the parents based on probabilities that favor a physically fit individual. The crossover operation creates offspring with pieces from both parents, and the solutions are more likely to be similar to the parents. Finally, the mutation process, which injects some discrepancy into the solutions occasionally, is how GA achieves its diversity. In the PSO algorithm, as the new swarm of particles is produced via the updates of the positions and velocity of each old individual, it can be said with confidence that they are much different from the old ones. The PSO algorithm converged so quickly, as the findings showed, due to the one-way influence of the best particle in the swarm over all other solutions in the population. This process limited the solution candidates and prevented further improvements. The DE algorithm improved the finding of new answers by ensuring that the best solution did not influence the other solutions in the population. In addition, the mutated vector was always a solution that did not come from the original population; therefore, the crossover operation in DE always took place between a population solution and a newly generated one. The further improvement of the DE algorithm was led by this process, unlike both PSO and GA algorithms, as the findings of this study show.

A laptop carried out the proposed calibration techniques with the following specifications: Lenovo ThinkPad T440s, CPU: Intel(R) Core TM i7-4600U @ 2.10GHz with 2 Core(s) and 4 Logical Processor(s), and RAM: 8.00 GB, and a 64-bit Operating System Win10. The DE, GA, PSO, HYBRID PSODE, and HYBRID GA calibration methods using MANE conclude the calibration process in 64.1, 45.4, 40.2, 55, and 59.5 seconds respectively, while using RMSE needed 102.7, 79.5, 51.2, 68,4 and 94,6 seconds respectively.

5 CONCLUSION AND FUTURE WORK

5.1 Conclusions

LUTI models require careful calibration to ensure their accuracy and reliability. Different calibration methods can be used to improve the accuracy of the model. Each method has its strengths and weaknesses, and the choice of calibration method depends on the nature of the problem and the data available. It is important to note that the calibration of LUTI models is an iterative process that requires multiple rounds of testing and refinement to achieve the desired level of accuracy. Overall, successful calibration is crucial for its effective use in urban planning and policymaking.

According to the literature reviewed, most of the existing LUTI calibration methods are semi-automated, using single-objective functions and local estimation techniques, and they suffer from the lack of a global estimation process. There is no standard approach to calibrating LUTI models or consensus on which objective function to use. However, their complexity makes the calibration of these tools costly, time-consuming, and challenging. To address these existing limitations, a novel LUTI model calibration technique benefiting from the capability of the differential evolution algorithm is presented in this study. This study is a step forward in developing a global and automated calibration approach for the LUTI models, which was the objective of prior investigations. By reformulating the TRANUS land use and activity module to make the calibration process more straightforward, we have contributed to this thesis. To accomplish this reformulation, we had to present the equations used in the computation of the land use and activity model module, utilizing the fundamental mathematical concepts underlying the microeconomic models employed and creating the necessary objective function using MANE and RMSE for the success of the optimization algorithms. First, the land-use model's most important parameters (elasticities, price factors, and shadow prices) were obtained through sensitivity analysis. Then, a DE algorithm was used to calibrate these parameters simultaneously to reach a global minimum using MANE and RMSE as multi-objective functions of both productions ($\boldsymbol{X^n j}$) and prices ($\boldsymbol{P^n j}$). The sensitivity analysis and the suggested calibration technique were tested on data from example C of the TRANUS model, and to be able to test our optimization approach, two optimization techniques (GA and PSO) and, further, the hybrid of GADE and PSODE were utilized to test the performance of this thesis proposed calibration approach using DE algorithm, with the usage of the same objective function method (MANE and RMSE).

Here is the summary of the research findings:

i. In terms of modeling the observed data, the calibration approach developed in this study using the DE algorithm demonstrated superior performance compared to both PSO and GA algorithms and the implemented version of the hybrid GADE and PSODE optimization techniques. The assessment was based on utilizing MANE and RMSE as multi-objective functions.
ii. DE-based optimization enables the incorporation of multiple parameters within the LUTI model calibration procedure.

iii. In this thesis, it is shown that MANE performs better than RMSE when employed as a multi-objective function, in all utilized optimization methods. MANE observed values are modeled with no (or low) discrepancy.

iv. Continuous improvement of the findings was proven in this study utilizing the DE calibration approach; however, the usage of both GA and PSO meant the results were stuck at a point, limiting future progress.

v. Because of DE operations in the hybrid optimization techniques (GADE and PSODE), optimization and discovery of new solutions continue, and they do not adhere to bounds, as is experienced in PSO and GA.

vi. When considering computational time, calibration techniques that employ MANE as a multi-objective function outperformed those using the RMSE as a multi-objective function.

vii. PSO-based calibration techniques demonstrated the fastest convergence time among the GA- and DE-based calibration techniques when employing both the MANE and the RMSE as multi-objective functions.

viii. In conclusion, when calibrating land-use model parameters, the recommended calibration technique using the DE algorithm and employing the MANE as a multi-objective function showed superior performance and faster convergence time than using the RMSE as a multi-objective function.

Overall, the choice of optimization method depends on the problem being solved and its characteristics. For example, DE and GA are often used for solving complex, nonlinear problems, while PSO is well-suited for problems with continuous variables. Hybrid methods like GADE and PSODE may be helpful for problems that require a combination of exploration and exploitation.

5.2 Future Works

i. Due to limited sources and data, this calibration approach is tested on a small example. However, it is recommended that the method be implemented on a more prominent and actual model for ideal results.

ii. To speed up the calibration of LUTI models with multiple parameters, parallel computation methods are crucial in reducing the time required.

iii. The current model only included production and prices as objective parameters to be modeled. In contrast, several parameters are considered and play a vital role in the land use and transportation models, specifically with sustainable mobility, such as mixed-use, multi-modality of traffic flows, population densities, accessibility, inclusiveness, connectivity to the existing communication network, and public transport.

iv. This study found that employing the DE algorithm as the calibration method for LUTI models yields remarkable results. However, there is still ample room for improvement, particularly when it comes to cases where the cost values (MANE or RMSE) approach zero, yet the modeled data (production and prices) for certain combinations of Sectors and Zones remain insufficiently close to the actual values. This area warrants further exploration in future research.

v. Another intriguing aspect that offers significant potential for improvement is the enhancement of the DE algorithm through the consideration of DE mutation strategies. Specifically, the DE algorithm calibration approach can benefit from further development and refinement.

vi. Incorporating hybridization, where different algorithms are combined, presents ample room and opportunities for enhancing the proposed calibration approach in this thesis. Exploring this avenue further could lead to developing a comprehensive and automated calibration approach for LUTI models on a global scale. It is an exciting area that could benefit from more research expansion and investigation.

»This page is intentionally blank.«

6 POVZETEK

6.1 Uvod

Vzajemno delovanje med rabo zemljišč in prometom je ključni vidik urbanega razvoja, ki je v ločenih postopkih načrtovanja pogosto spregledan, kar povzroča težave, kot je širjenje mest. Namen modelov interakcije med rabo zemljišč in prometom (LUTI) je napovedati zapletene odnose med gospodarsko rastjo, povpraševanjem po prometu in vzorci rabe zemljišč. Vendar je bila zanesljivost teh modelov vprašljiva, zato so potrebni zanesljivi postopki umerjanja in potrjevanja.

Motivacija za to raziskavo je ključna vloga rabe mestnih zemljišč in prometne infrastrukture pri razvoju mesta. Modeli LUTI, ki se razvijajo od petdesetih let prejšnjega stoletja, združujejo ekonometrijo, demografijo in prometno inženirstvo za simulacijo posledic različnih scenarijev načrtovanja na rabo zemljišč in prometne vzorce. Pomanjkanje splošno sprejetega postopka ocenjevanja in globalnega kalibracijskega pristopa je oviralo splošno zaupanje v rezultate modelov LUTI.

Cilji raziskave se osredotočajo na reševanje vprašanj zaupanja, povezanih z modeli LUTI, z razvojem samodejnega in globalnega kalibracijskega pristopa. Študija uporablja algoritem diferencialne evolucije (DE), ki je zmogljiv evolucijski algoritem, za oceno učinkovitosti kalibracije pa uporablja statistične metrike, kot sta korenska srednja kvadratna napaka (RMSE) in srednja absolutna normalizirana napaka (MANE). Za testiranje predlaganega kalibracijskega pristopa je izbran modul rabe tal in dejavnosti modela TRANUS LUTI.

Hipoteza, preverjena v tej raziskavi, je, da bo uporaba algoritma DE izboljšala kalibracijo modelov LUTI.

Disertacija vsebuje pregled modelov LUTI, tehnik umerjanja, metod optimizacije ter poseben poudarek na algoritmu DE in modelu TRANUS LUTI. Podrobno je opisan predlagani samodejni in globalni pristop kalibracije, ki vključuje analizo občutljivosti, opredelitev ciljnih funkcij ter prilagoditev tehnik kalibracije in optimizacije. Študija se zaključi s predstavitvijo rezultatov kalibracije, razpravami in primerjavami različnih tehnik ter predstavi omejitve in morebitna področja za prihodnje raziskave.

6.2 Ozadje

6.2.1 LUTI modeli

Modeli interakcije med rabo zemljišč in prometom (LUTI) imajo ključno vlogo pri urbanističnem načrtovanju, saj analizirajo zapletene povezave med vzorci rabe zemljišč in prometnimi sistemi v mestih. Ti modeli simulirajo vpliv sprememb rabe zemljišč in prometne infrastrukture na potovalno vedenje, dostopnost in splošno prostorsko strukturo mestnih območij. Modeli LUTI združujejo matematične, statistične in simulacijske tehnike, pri čemer uporabljajo različne vire podatkov, kot so podatki iz popisa prebivalstva, zemljevidi rabe zemljišč in podatki o prometnem omrežju.

Raziskana je zgodovina in uporaba modelov LUTI s poudarkom na njihovi zmožnosti napovedovanja demografskih in gospodarskih kazalnikov za dejavnosti na zemljiščih ter prometnih vzorcev v prometnih omrežjih. Povratna zanka v modelih LUTI omogoča upoštevanje vzajemnega vpliva prometnih politik na vzorce rabe zemljišč in obratno, zaradi česar so dragocena orodja za oblikovalce politik.

Predstavljena je klasifikacija modelov LUTI na podlagi zgodovinskega razvoja, ki jih deli na prvo, drugo in tretjo generacijo. Prva generacija, ki sega v šestdeseta in sedemdeseta leta prejšnjega stoletja, vključuje interakcijske modele, modele matematičnega programiranja, ki temeljijo na optimizaciji, in modele, ki temeljijo na matrikah input/output. Modeli druge generacije, ki so se pojavili v osemdesetih in devetdesetih letih prejšnjega stoletja, temeljijo na McFaddenovem delu o teoriji naključne koristnosti. Modeli tretje generacije, ki so nastali konec devetdesetih let prejšnjega stoletja, so zelo razčlenjeni in dinamični, kot je URBANSIM.

Izzivi pri modelih LUTI vključujejo uporabo nekoliko zastarelih pristopov pri simulaciji prometnih podsistemov, kot je tradicionalna štiristopenjska zaporedna metoda. Kljub napredku nobena generacija modelov ni popolnoma nadomestila drugih, zato se predlaga sprejetje sodobnejših, endogenih ali eksogenih modelov.

Glavni cilj modelov LUTI je opisati zapleteno medsebojno vplivanje med rabo zemljišč in prometom v urbanih okoljih. Njihov razvoj, ki ga spodbujajo vse večje računalniške zmogljivosti in teoretični preboji, je namenjen zagotavljanju bolj realističnih predstavitev in boljših orodij za odločanje pri dolgoročnem načrtovanju v mestih in regijah. Tehnike umerjanja in potrjevanja ostajajo ključne za vzpostavitev in pojasnitev operativne zmogljivosti modelov LUTI.

6.2.2 TRANUS

Model TRANUS, ki ga je zasnoval Tomás de la Barra, velja za široko uporabljan odprtokodni model interakcije med rabo zemljišč in transportom (LUTI). Združuje modele rabe zemljišč in dejavnosti s prometnim modelom, ki deluje kot input-output model, ki tesno povezuje vse komponente urbanih ali regionalnih sistemov. TRANUS je posebej razvit za simulacijo posledic različnih projektov in politik v mestih in regijah, vrednotenje rezultatov iz socialno-ekonomskih, proračunskih in okoljskih perspektiv. Model deluje na podlagi makroekonomskega ravnovesja, pri čemer obravnavano območje razdeli na prostorske cone in gospodarske sektorje. TRANUS sestavljata dva modula: modul rabe zemljišč in dejavnosti ter modul transporta. Prvi simulira prostorski ekonomski sistem z analizo lokacij dejavnosti in odnosov med gospodarskimi sektorji, medtem ko drugi izračuna uporabo prometnega omrežja in s tem povezano neuporabnost. Oba modula uporabljata teorijo naključne uporabnosti in uporabljata logit modele diskretne izbire za različne oznake.

TRANUS doseže ravnotežje s ponavljajočim se postopkom, pri čemer upošteva dejavnike, kot so lokacija dejavnosti, raba zemlje in neuporabnost prevoza. Model uporablja vhodno-izhodni okvir z neprekinjenimi poteki, dokler ni doseženo splošno ravnotežno stanje. Modul za rabo zemljišč in dejavnosti izračunava rezultate, porabo in povpraševanje po tokovih, medtem ko modul za transport dodeljuje potovalne tokove na podlagi povpraševanja.

Uporabljen je standardni okvir input-output modela, ki razlikuje med prevoznimi in neprenosnimi sektorji. Prenosni sektorji proizvajajo tokove, medtem ko se neprenosni sektorji porabljajo le tam, kjer so ustvarjeni. Gospodarski sektorji so razvrščeni v zemljišča ali površine, gospodinjstva in industrije, od katerih je vsak predstavljen z različnimi parametri in funkcijami. Parametri elastičnosti, sektorska teža, parametri disperzije in drugi prispevajo k modeliranju obnašanja gospodarskih subjektov.

Matematične enačbe TRANUS-a vključujejo značilnosti, kot so eksogena in inducirana proizvodnja, eksogeno in inducirano povpraševanje, stroški potrošnje, proizvodni stroški in dodana vrednost. Te značilnosti igrajo ključno vlogo pri analizi občutljivosti med predlaganim pristopom kalibracije. Cilj TRANUS-a je zagotoviti celovito orodje za odločanje pri dolgoročnem načrtovanju, ki vključuje zapletene interakcije med rabo zemljišč in prevozom v urbanem kontekstu.

6.2.3 Kalibracija modela LUTI

Umerjanje modelov interakcije med rabo tal in prometom (LUTI) je zaradi negotovosti, ki je neločljivo povezana s temi numeričnimi modeli in izhaja iz teoretičnih predpostavk in kakovosti podatkov, zelo pomemben proces. Kalibracija vključuje ocenjevanje in prilagajanje parametrov modela za zmanjšanje razlik med dejanskimi in modeliranimi podatki. Kljub izzivom, ki jih predstavljajo teoretična, metodološka in praktična vprašanja, je kalibracija bistvena za potrjevanje rezultatov simulacij, zlasti v majhnih merilih.

Modelu TRANUS LUTI uporablja dva pristopa h kalibraciji: ad hoc ekonometrične metode ter metode poskusov in napak. Pomanjkanje kalibracije lahko privede do netočnih odločitev v postopku načrtovanja, zato je kalibracija ključnega pomena za izboljšanje natančnosti simulacije. Postopek umerjanja vključuje določitev parametrov modela z uporabo podatkov, predhodnega znanja ali predvidenih rezultatov modela. Vendar pomanjkanje natančnih in učinkovitih načinov za kalibracijo parametrov, zlasti pri obsežnih modelih, ostaja izziv.

Raziskani so bili različni pristopi k umerjanju modela LUTI. Pogosto se uporabljajo optimizacijske metode, kot sta optimizacija z največjo verjetnostjo in optimizacija s pomočjo roja delcev (PSO). Poleg tega so raziskovalci razvili polavtomatske pristope kalibracije za določene modele LUTI, kot so MEPLAN, PECAS, Pirandello, UrbanSim in ITLUP. Ti pristopi vključujejo tehnike, kot so optimizacija po metodi najmanjših kvadratov, spuščanje po naklonu in analiza občutljivosti s simulacijo Monte Carlo. Modelu MUSSA za umerjanje uporablja ekonometrične pristope, model TRANUS pa je bil uporabljen na različnih lokacijah po svetu, vključno z Brazilijo in Francijo, pri čemer se tehnike umerjanja razlikujejo od ad hoc metod do ekonometričnih metodologij. V nekaterih študijah so bile predlagane verjetnostne metodologije preverjanja in preučeno širjenje negotovosti med postopkom umerjanja z uporabo metode Monte Carlo.

Kljub velikemu napredku na področju ekonometrije in optimizacije se pri umerjanju dinamičnih ali kvazidinamičnih modelov še vedno pojavljajo izzivi. Tehnike umerjanja v objavljenih člankih pogosto niso v celoti opisane, zato obstaja potreba po globalnih in samodejnih strategijah umerjanja za modele LUTI. Na splošno je umerjanje modelov LUTI zapletena in ključna naloga, ki zahteva skrbno upoštevanje različnih dejavnikov, da se povečata natančnost in zanesljivost rezultatov simulacij.

6.2.4 Optimizacija

Optimizacija je široko področje uporabne matematike z različnimi podpodročji, ki se ukvarjajo s problemi s posebnimi značilnostmi, da bi našli učinkovite rešitve. V pregledu so obravnavane trenutne spremembe in prihodnji trendi na področju optimizacijskih tehnik.
Pregled optimizacije: Optimizacija se osredotoča na določanje skrajne vrednosti funkcije na določenem področju ob upoštevanju več vrednosti spremenljivk. Splošna oblika optimizacijskega problema vključuje optimizacijo ciljnih ali stroškovnih funkcij ob upoštevanju enakosti, neenakosti in stranskih omejitev.
Razvrstitev pristopov k optimizaciji: Optimizacijski pristopi se delijo na lokalne in globalne algoritme.
Lokalna optimizacija: Lokalna optimizacija je namenjena iskanju enega od ekstremov funkcije. Algoritmi, ki temeljijo na gradientu in se pogosto uporabljajo pri lokalni optimizaciji, imajo omejitve, vključno z iskanjem le lokalnih optimumov in težavami pri diskretnih optimizacijskih problemih.
Globalna optimizacija: Za razliko od lokalne optimizacije je cilj globalne optimizacije določiti globalno najmanjšo vrednost funkcije ne glede na začetni položaj. Deterministična globalna optimizacija zagotavlja dragocene informacije v nekonveksnih situacijah z več optimami. Algoritmi globalne optimizacije imajo v primerjavi z lokalnimi algoritmi večjo verjetnost, da odkrijejo globalne ali skoraj globalne optimalne vrednosti. Tehnike so razdeljene na stohastično iskanje (npr. evolucijsko računalništvo) ali deterministične algoritme.
Avtomatizirana strategija globalne optimizacije z algoritmom DE: V ospredju je razvoj avtomatizirane in globalne optimizacijske strategije za kalibracijo modela LUTI z algoritmom diferencialne evolucije (DE), ki je član družine evolucijskih algoritmov. Stohastično-hevristične metode globalne optimizacije, kot je DE, so bile uporabljene za reševanje večjih problemov, čeprav brez zagotovil o rešitvah ali konvergenčnem obnašanju.
Ta povzetek vsebuje poglobljen pregled optimizacije s poudarkom na razlikovanju med lokalnimi in globalnimi optimizacijskimi tehnikami ter izpostavlja uporabo algoritma diferencialne evolucije v okviru kalibracije modela LUTI.

6.2.4.1 Večnamenski pregled optimizacije

Končni cilj optimizacijskih problemov je optimizacija ene ali več ciljnih funkcij. Medtem ko se večina metod osredotoča na eno samo ciljno funkcijo, se večnamenska optimizacija ukvarja z optimalnim kompromisom med dvema ali več povezanimi cilji, kar je pogosto v kompleksnih inženirskih aplikacijah. Evolucijski algoritmi (EA), zlasti genetski algoritmi (GA), optimizacija z rojem delcev (PSO) in diferencialna evolucija (DE), so se izkazali za učinkovite pri reševanju izzivov večobjektne optimizacije.

Evolucijsko računanje vključuje iskanje rešitev z ocenjevanjem vrednosti fitnesa, izbiro boljših rešitev ter generiranje novih kandidatov s pomočjo evolucijskih in selekcijskih procesov. EC se zgleduje po naravni selekciji in genetiki ter uporablja izraze, kot so posamezniki, populacije, genomi, kromosomi in potomci. Zaradi svoje učinkovitosti pri reševanju zapletenih problemov optimizacije se pogosto uporablja.

EAs, ki temeljijo na načelih biološke evolucije, delajo z naborom rešitev, ki se posodabljajo s postopki, kot so križanje, mutacija in selekcija. GA, PSO in DE so priljubljeni EA za večpredmetno optimizacijo. Obravnavajo izzive kombinatorične optimizacije, pri čemer je GA dobro uveljavljen, PSO in DE pa pridobivata pozornost zaradi uporabe pri večpredmetni optimizaciji.

GA je evolucijski algoritem, ki ga je zasnoval John Holland in posnema genetske in evolucijske koncepte. GA uporabljajo binarne nize za predstavitev kromosomov in uporabljajo genetske operatorje, kot so križanje, mutacija in naključna izbira. Odlični so pri reševanju realnih, nejasnih in zapletenih optimizacijskih vprašanj ter primerni za diskretne in hrupne prostore.

PSO, ki ga je navdihnilo obnašanje rojev v naravi, uporablja delce za predstavitev potencialnih rešitev v iskalnem prostoru. Vsak delec ima spomin, ki mu omogoča, da se vrne k prej znanim rešitvam. PSO se odlikuje po preprostem izvajanju in je boljši od drugih algoritmov glede uspešnosti, kakovosti rešitev in hitrosti konvergence.

Diferencialne Evolucije (DE):

Algoritem DE, ki sta jo uvedla Storn in Price, je stohastični evolucijski algoritem, ki temelji na populaciji in je namenjen reševanju problemov v zveznih prostorih. DE je znan po svoji preprostosti, učinkovitosti in uspešnosti pri reševanju optimizacijskih izzivov, zlasti zaradi omejenih kontrolnih parametrov. Za razliko od genetskih algoritmov (GA), DE vključuje samoprilagodljivo mutacijsko shemo, ki zagotavlja prednosti, kot so učinkovita uporaba pomnilnika, manjša računska zahtevnost in hitrejša konvergenca.

DE odlikujejo preprostost, enostavnost uporabe, hitrost in velika verjetnost odkritja globalnih optimalnih rešitev. Njegova vsestranskost se razteza na aplikacije v celoštevilski in diskretni optimizaciji, nelinearni optimizaciji z omejitvami in kazenskimi funkcijami ter večmodalnih, večpredmetnih, omejenih in dinamičnih modelih.

DE deluje kot metahevristična tehnika, ki temelji na populaciji in uporablja populacijo NP posameznikov, ki jih predstavljajo D-dimenzionalne odločitvene spremenljivke. Algoritem sestavljajo trije glavni operatorji: mutacijo, križanje (rekombinacijo) in izbiro.

- Inicializacija: DE se začne s populacijo posameznikov NP, od katerih vsakega označuje vektor D-dimenzionalnih odločitvenih spremenljivk.

- Mutacija: Pri mutaciji se na podlagi ciljnega vektorja ustvari nov vektor potomcev. DE uvaja strategije mutacije, označene z DE/x/y/z, kjer x predstavlja vrsto ciljnega vektorja ("naključni" ali "najboljši"), y označuje število uporabljenih vektorjev razlike, z pa označuje operator rekombinacije (binomski ali eksponentni).
- Po mutaciji križanje ustvari poskusni vektor z uporabo ciljnega in donorskega vektorja. Parameter stopnje križanja, Cr, nadzoruje stopnjo motenj osnovnega (ciljnega) vektorja. V DE se običajno uporabljata tehniki enakomernega (binomskega) in eksponentnega križanja.
- Izbor: Izbira z uporabo pohlepnega pristopa primerja primernost poskusnega vektorja s primernostjo ciljnega vektorja. Vektor z najboljšo primernostjo se izbere kot novi član populacije in v naslednji generaciji nadomesti ciljni vektor.

DE je pridobil veliko pozornosti in je bil uspešno uporabljen za različne optimizacijske probleme. Učenjaki so raziskali različne strategije mutacije, nastavitve prilagodljivega nadzora in hibridizacijske pristope, da bi izboljšali učinkovitost DE. Predlagane so bile prilagodljive različice DE, združevanje DE z drugimi algoritmi in uporaba samoprilagodljivih kontrolnih nastavitev parametrov.

Najnovejši dosežki vključujejo tehnike za izboljšanje raznolikosti populacije, strategije za reševanje situacij stagnacije, sisteme z več mutacijami, prilagodljive operatorje mutacij za razvrščanje za omejene probleme in dvostopenjske pristope k podpopulacijam. Raziskovalci so se osredotočili tudi na skalabilnost za izzive optimizacije z velikim številom podatkov, uporabo kooperativne koevolucije in nadgradnjo DE za omejeno optimizacijo z več metodami lokalnega iskanja.

Različni raziskovalci so predlagali prilagodljive kontrolne nastavitve DE, predstavljeni pa so bili tudi podrobni pregledi in taksonomije različic DE. Uporaba posplošenega faktorja skaliranja, sintetičnih mutacijskih operatorjev s prilagajanjem parametrov in vključevanje kovariančnih matrik prikazujejo nenehno evolucijo in izpopolnjevanje algoritma DE.

Če povzamemo, diferencialna evolucija ostaja močno in vsestransko orodje za optimizacijo, ki se nenehno razvija z inovativnimi strategijami, prilagodljivim nadzorom in tehnikami hibridizacije za reševanje različnih in kompleksnih izzivov optimizacije. Raziskovalci še naprej raziskujejo njene zmogljivosti in izboljšujejo njeno učinkovitost za različna področja uporabe.

6.3 Metodologija

Raziskovalna metodologija za kalibracijo modela LUTI (Land-Use Transportation Integration) vključuje celoten postopek oblikovanja modela, vključno z določitvijo gospodarskih sektorjev, pridobivanjem podatkov in regionalnim določanjem območij. Kalibracija se v tem kontekstu nanaša na ocenjevanje parametrov modela za ponovitev podatkov iz prejšnjega leta v raziskovalni regiji. Ta dolgotrajen postopek običajno izvajajo strokovnjaki in lahko traja več mesecev.

V primeru modela TRANUS, modela za regije z N sektorji in M območji, kalibracija vključuje posodobitev cen v senci, da se proizvodnja ujema z opazovanimi podatki. Cene v senci služijo kot korekcijski pogoji, ki izravnavajo vidike, ki se ne odražajo v modelu.

Predlagan je nov pristop kalibracije, ki uporablja algoritem diferencialne evolucije za samodejno in globalno ocenjevanje parametrov. Namen tega pristopa je nadomestiti trenutni zaporedni postopek kalibracije programa TRANUS in oceniti uspešnost z uporabo večobjektivnih funkcij (MANE in RMSE). Opravljena je primerjava z genetskim algoritmom (GA) in optimizacijo s pomočjo roja delcev (PSO), uspešnost optimizacije pa je izboljšana s hibridnim PSODE in hibridnim GADE.

V fazi umerjanja se izvede analiza občutljivosti, da se oceni vpliv vhodnih parametrov na rezultate. Globalna analiza občutljivosti z uporabo posplošenih Sobolovih indeksov opredeli vplivne parametre. Program TRANUS je kodiran v Pythonu, Sobolovi indeksi pa so ocenjeni z uporabo programa SALIB, pri čemer se kot funkcija napake upošteva srednja absolutna normalizirana napaka (MANE).

Za količinsko opredelitev napake med dejanskimi in simuliranimi rezultati sta uporabljeni objektivni funkciji, in sicer korenska srednja kvadratna napaka (RMSE) in srednja absolutna normalizirana napaka (MANE). Te večobjektivne funkcije se pogosto uporabljajo na različnih raziskovalnih področjih, vključno z meteorologijo, kakovostjo zraka, podnebjem, prometom in transportom.

Umerjanje modelov LUTI je ključnega pomena za razvoj uporabniku prijaznih splošnih modelov z zanesljivimi rezultati za nosilce odločanja. Za kalibracijo parametrov modelov so bile uporabljene različne metode optimizacije, kot so genetski algoritem, optimizacija roja delcev, ocenjevanje največjega verjetja in diferencialna evolucija (DE). Algoritem DE je za predlagani pristop kalibracije izbran zaradi svojih prednosti, vključno z možnostjo samodejnega in globalnega ocenjevanja.

Algoritem diferencialne evolucije je izbran zaradi svoje preprostosti in enostavnosti izvajanja. Algoritem deluje v n-razsežnem iskalnem prostoru in uporablja postopke, kot so inicializacija, mutacija, križanje in izbira. V fazi mutacije se uporabljajo posebne tehnike za izrezovanje mutantnih vrednosti na podlagi zgornjih in spodnjih mej želenih parametrov.

Nastavitve parametrov za algoritem DE: Vključevanje parametrov, kot so mutacijski faktor (μF), stopnja križanja (CR) in velikost populacije (NP), pomembno vpliva na učinkovitost algoritma DE. V študiji je uporabljeno samodejno iskanje optimalnih kombinacij v določenih razponih. Prejšnje študije priporočajo vrednosti NP znotraj [3, 8] *D, μF=0,6 in CR znotraj [0,3, 0,9].

Uporabljene so različne strategije mutacije, pet običajnih strategij pa je modificiranih na podlagi primera modela LUTI. Študija poudarja, da je strategija mutacije ključna za doseganje najboljših rezultatov v algoritmu DE.

GA vključuje naključno izbrano populacijo, oceno primernosti, izbiro staršev, križanje in mutacijo. Nastavitve parametrov, zlasti verjetnost križanja (Cr) in stopnje mutacije (mu), se ocenijo v določenem območju.

PSO je algoritem, ki temelji na populaciji in uporablja delce, ki predstavljajo potencialne rešitve. Parametri, kot so osebni in globalni učni koeficienti ter utež inercije, se ocenijo za optimalno delovanje.

Hibridne strategije (PSODE in GADE): Hibridizacija vključuje združevanje operatorjev in različic različnih tehnik optimizacije. PSODE (hibrid PSO in DE) in GADE (hibrid GA in DE) sta uvedena za

izboljšanje kalibracijskega pristopa. Nastavitve parametrov za oba hibridna algoritma so ovrednotene s polavtomatskim postopkom.
Shema poteka kalibracijskega pristopa in grafični uporabniški vmesnik predlagata uporabniku prijazen pristop za optimizacijo modela LUTI. Merila za zaustavitev za optimizacijske tehnike so opredeljena na podlagi konvergence in natančnosti.

6.4 Rezultati in razprava

Algoritem diferencialne evolucije (DE) se izkaže za najuspešnejšo tehniko optimizacije. Prekaša genetski algoritem (GA) in optimizacijo s pomočjo roja delcev (PSO) v smislu metrik povprečne absolutne normalizirane napake (MANE) in korenske povprečne kvadratne napake (RMSE).
Izboljšanje hibridnih strategij: Medtem ko DE izstopa, študija vključuje hibridne strategije (GADE in PSODE), pri čemer je prikazana njuna učinkovitost pri izboljšanju rezultatov kalibracije v primerjavi s samostojnima GA in PSO. Hibridni pristopi so pokazali opazen napredek pri doseganju bolje uglašenih parametrov.
Algoritem DE dosledno dokazuje svojo sposobnost konvergence k optimalnim rešitvam. Vrednosti MANE skoraj dosežejo ničlo, kar kaže na tesno ujemanje z opazovanimi podatki. Iterativna narava algoritma DE mu omogoča, da nenehno izboljšuje rezultate kalibracije v več iteracijah.
Čeprav je pri optimizaciji s pomočjo roja delcev (PSO) konvergenca hitra in optimalno vrednost MANE doseže v samo 110 iteracijah, se le stežka še izboljša, kar kaže na morebitne omejitve pri raziskovanju prostora rešitev.
V študiji sta za oceno tehnik umerjanja uporabljeni večobjektivni funkciji MANE in RMSE. DE z uporabo MANE doseže natančnejše vrednosti parametrov, ki se natančno ujemajo z opazovanimi podatki o proizvodnji in cenah. Nasprotno pa DE z uporabo RMSE povzroči večje neskladje med modeliranimi in opazovanimi vrednostmi, kar poudarja pomen izbire ustreznih ciljnih funkcij.
Predstavljeni so rezultati kalibracije za DE, GA in PSO, ki kažejo optimizirane vrednosti parametrov, povezanih z gospodarskimi sektorji in rabo zemljišč. DE dosledno prekaša GA in PSO, kar kaže na njeno sposobnost učinkovitejšega natančnega prilagajanja vrednosti parametrov.
V študiji so primerjane modelirane vrednosti, pridobljene z umerjanjem, z opazovanimi podatki iz modela TRANUS. DE, zlasti pri uporabi MANE, daje vrednosti parametrov, ki so tesno usklajene z dejanskimi podatki modela TRANUS. Primerjave poudarjajo učinkovitost algoritma DE pri zajemanju kompleksnosti modela rabe tal.
Podrobna primerjava hibridnih strategij GADE in PSODE pokaže razlike v njuni učinkovitosti. GADE, zlasti pri uporabi MANE, daje dobro prilegajoče se vrednosti v primerjavi z opazovanimi rezultati TRANUS. Študija poudarja, da je treba oba hibridna modela skrbno ovrednotiti in pri tem upoštevati dejavnike, kot sta prileganje dejanskim podatkom in morebitno pretirano prileganje.

Primerjava učinkovitosti: Primerjava časov obdelave za DE, GA in PSO. PSO je najhitrejši, saj postopek umerjanja opravi v najkrajšem času. To učinkovitost je mogoče pripisati enosmernemu vplivu najboljšega delca v roju, ki omejuje kandidate za rešitev in preprečuje nadaljnje izboljšave.
Če povzamemo, je v razdelku Rezultati in razprava podana celovita analiza rezultatov kalibracije, pri čemer so poudarjene prednosti algoritma DE, prednosti hibridnih strategij in pomen izbire ustreznih ciljnih funkcij. Študija poudarja praktično uporabnost predlaganega kalibracijskega pristopa, zlasti z uporabniku prijaznim grafičnim uporabniškim vmesnikom, ter ponuja vpogled v prihodnje raziskave in izbiro optimizacijske strategije.

6.5 Zaključek in Prihodnje delo

Umerjanje modelov interakcije med rabo zemljišč in prometom (Land-Use Transport Interaction - LUTI) je ključnega pomena za zagotavljanje njihove natančnosti pri urbanističnem načrtovanju. Obstoječe metode so polavtomatske, nimajo globalnega postopka ocenjevanja in se soočajo z izzivi zaradi kompleksne in dolgotrajne narave modelov LUTI. Ta študija uvaja novo tehniko kalibracije z uporabo algoritma diferencialne evolucije (DE), katere cilj je globalni in samodejni pristop.
Predlagana tehnika kalibracije DE presega tradicionalne algoritme (PSO, GA) in hibridne strategije (GADE, PSODE) v smislu modeliranja opazovanih podatkov z uporabo večobjektivnih funkcij MANE (Mean Absolute Normalized Error) in RMSE (Root Mean Square Error). b) Kalibracija več parametrov: DE omogoča hkratno kalibracijo več parametrov v modelu LUTI, kar omogoča celovit pristop k optimizaciji. c) Superiornost MANE: MANE kot večpredmetna funkcija dosledno presega RMSE pri vseh metodah optimizacije, kar kaže na njeno učinkovitost pri doseganju bolje kalibriranih rezultatov. d) Nenehno izboljševanje z DE: DE kaže nenehno izboljševanje, medtem ko GA in PSO običajno ostajata pri mejah parametrov. Hibridne različice (GADE, PSODE) z vključevanjem DE kažejo stalno optimizacijo brez mejnih omejitev. e) Računska učinkovitost: Kalibracijske tehnike, ki uporabljajo MANE, kažejo hitrejšo konvergenco v primerjavi s tistimi, ki uporabljajo RMSE. Kalibracija, ki temelji na PSO, izkazuje najboljši čas konvergence med tehnikami, ki temeljijo na PSO, GA in DE ter uporabljajo tako MANE kot RMSE.
Predlagana tehnika kalibracije DE, ki uporablja MANE kot večpredmetno funkcijo, je boljša od drugih metod v smislu učinkovitosti (razmerje med modeliranim in opazovanim) in časa konvergence. Izbira optimizacijske metode je odvisna od specifičnih značilnosti problema, pri čemer sta DE in GA primerna za kompleksne, nelinearne probleme, PSO pa za probleme z zveznimi spremenljivkami. Hibridne metode, kot sta GADE in PSODE, ponujajo ravnovesje med raziskovanjem in izkoriščanjem.
Zaradi omejenih podatkov je pristop kalibracije preizkušen na majhnem primeru, pri čemer se priporoča uporaba na večjih, resničnih modelih. Ugotovljene so nadaljnje izboljšave kalibracije DE, zlasti kadar se vrednosti stroškov približujejo ničli, vendar modelirani podatki še vedno odstopajo od dejanskih vrednosti. Izboljšanje mutacijskih strategij pri kalibraciji DE in raziskovanje hibridizacije z različnimi

algoritmi ponujata priložnosti za prihodnje raziskave. Cilj je napredovati v smeri globalnega in samodejnega pristopa kalibracije za modele LUTI, ki bo obravnaval zapletenost in izboljšal natančnost.

7 REFERENCES (IEEE)

[1] M. Wegener, "OVERVIEW OF LAND-USE TRANSPORT MODELS," *Transport Geography and Spatial Systems. Handbook 5 of the Handbook in Transport. Pergamon/Elsevier Science, Kidlington, UK*, vol. 9, no. 5, pp. 127–146, 2004.

[2] L. Hellman, "' Learning from Las Vegas'.," *Built Environ*, vol. 8, no. 4, pp. 267–271, 1982, doi: 10.1002/sdr.

[3] J. D. Hunt, D. S. Kriger, and E. J. Miller, "Current operational urban land-use-transport modelling frameworks: A review," *Transp Rev*, vol. 25, no. 3, pp. 329–376, 2005, doi: 10.1080/0144164052000336470.

[4] E. Feitelson, D. Felsenstein, E. Razin, and E. Stern, "Assessing land use plan implementation: Bridging the performance-conformance divide," *Land use policy*, vol. 61, pp. 251–264, 2017, doi: 10.1016/j.landusepol.2016.11.017.

[5] P. Pfaffenbichler, G. Emberger, and S. Shepherd, "The integrated dynamic land use and transport model MARS," *Netw Spat Econ*, vol. 8, pp. 183–200, 2008, doi: DOI 10.1007/s11067-007-9050-7.

[6] R. E. Klosterman, "Simple and complex models," *Environ Plann B Plann Des*, vol. 39, no. 1, pp. 1–6, 2012, doi: 10.1068/b38155.

[7] M. Batty, "Urban Modeling," *International Encyclopedia of Human Geography*, pp. 51–58, 2009, doi: 10.1016/B978-008044910-4.01092-0.

[8] N. Gaud, C. Boittin, and V. Hilaire, "Algorithmic calibration of Land-Use and Transport Integrated models," 2014.

[9] R. Storn and K. Price, "Differential Evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997, doi: 10.1023/A:1008202821328.

[10] T. de la Barra, "Integrated Land Use and Transport Modelling: Decision chains and hierarchies," *Cambridge University Press*, p. 194, 1989.

[11] de la Barra, "Mathematical description of TRANUS," *Technical report, Modelistica, Caracas, Venezuela.*, 1999.

[12] P. M. Torrens, "How land-use transportation models work," *Complexity*, p. 75, 2000.

[13] P. Waddell, G. F. Ulfarsson, J. P. Franklin, and J. Lobb, "Incorporating land use in metropolitan transportation planning," *Transp Res Part A Policy Pract*, vol. 41, no. 5, pp. 382–410, 2007, doi: 10.1016/j.tra.2006.09.008.

[14] M. Iacono, D. Levinson, and A. El-Geneidy, "Models of Transportation and Land Use Change: A Guide to the Territory," *J Plan Lit*, vol. 22, no. 4, pp. 323–340, 2008, doi: 10.1177/0885412207314010.

[15] P. Coppola, Á. Ibeas, L. dell'Olio, and R. Cordera, "LUTI Model for the Metropolitan Area of Santander," *J Urban Plan Dev*, vol. 139, no. 3, pp. 153–165, 2013, doi: 10.1061/(asce)up.1943-5444.0000146.

[16] I. S. Lowry, "A model of metropolis," *Rand Corporation*. pp. 1–150, 1964.

[17] R. B. Andrews, “Mechanics of the Urban Economic Base: Historical Development of the Base Concept,” *Land Econ*, vol. 29, no. 2, p. 161, 1953, doi: 10.2307/3144408.

[18] A. G. Wilson, “Entropy in urban and regional modelling,” *Entropy in Urban and Regional Modelling*, pp. 1–166, 1970, doi: 10.4324/9780203142608.

[19] L. Use, T. Environment, M. P. Description, L. Marquez, and N. Smith, “TOPAZ 2000 THEORY & APPLICATION by,” no. September, pp. 1–15, 2000.

[20] W. W. Leontief, *Input-Output Economics, 2nd ed.* 1986.

[21] J. E. Abraham and J. D. Hunt, “SEMI-AUTOMATED CALIBRATION OF THE MEPLAN MODEL OF SACRAMENTO,” pp. 1–28, 1999.

[22] J. E. Abraham, “Parameter estimation in urban models: Theory and application to a land use transport interaction model of the Sacramento, California region,” *ProQuest Dissertations and Theses*, p. 188, 2000, doi: 10.11575/PRISM/20470.

[23] D. McFadden, “Conditional logit analysis of qualitative choice behaviour,” in *Frontiers in Econometrics*, P. Zarembka, Ed., Academic Press New York, 1973, pp. 105–142. doi: 10.1080/07373937.2014.997882.

[24] A. Anas, “Discrete choice theory, information theory and the multinomial logit and gravity models,” *Transportation Research Part B*, vol. 17, no. 1, pp. 13–23, 1983, doi: 10.1016/0191-2615(83)90023-1.

[25] F. Martínez and P. Donoso, “MUSSA : a behavioural land use equilibrium model with location externalities, planning regulations and pricing policies .,” *7th International Conference on Computers and Urban Planning and Urban Management (CUPUM 2001)*, pp. 1–14, 2010, [Online]. Available: http://www.mussa.cl/E_descargas.html

[26] P. Waddell, “UrbanSim: Modeling urban development for land use, transportation, and environmental planning,” *Journal of the American Planning Association*, vol. 68, no. 3. pp. 297–314, 2007. doi: 10.1080/01944360208976274.

[27] M. Wegener *et al.*, “Land Use and Transportation Modeling,” *Handbook of transport geography and spatial systems*, 2004, doi: 10.1061/9780784479926.030.

[28] P. Dutta, M. Saujot, E. Arnaud, B. Lef, and E. Prados, “Uncertainty Propagation and Sensitivity Analysis During Calibration of an Integrated Land Use and Transport Model,” *International Journal of Civil and Environmental Engineering*, vol. 6, no. 6, pp. 121–129, 2012.

[29] N. Pupier, “Construction and Calibration of a Land-Use and Transport Interaction Model of a Brazilian City,” p. 54, 2013.

[30] B. Lefèvre, “Long-term energy consumptions of urban transportation: A prospective simulation of ‘transport-land uses’ policies in Bangalore,” *Energy Policy*, vol. 37, no. 3, pp. 940–953, 2009, doi: 10.1016/j.enpol.2008.10.036.

[31] T. de la Barra, “TRANUS : Integrated Land Use and Transport Modeling System,” p. 23, 2011, [Online]. Available: http://www.tranus.com/

[32] T. de la Barra, “TRANUS Tutorial : Step-by-step examples TRANUS Tutorial : Step-by-step examples,” no. May, 2017.

[33] M. Saujot, M. de Lapparent, E. Arnaud, and E. Prados, “To make LUTI models operational tools for planning,” *The 14th International Conference on Computers in Urban Planning and Urban Management*, 2015, [Online]. Available: http://web.mit.edu/cron/project/CUPUM2015/proceedings/Content/pss/312_saujot_h.pdf

[34] C. Boittin, N. Gaud, V. Hilaire, and D. Meignan, “Particle swarm for calibration of land-use and transport integrated models,” *CUPUM 2015 - 14th International Conference on Computers in Urban Planning and Urban Management*, 2015.

[35] P. Bonnel *et al.*, “A survey on the calibration and validation of integrated land use and transportation models To cite this version :,” *Symposium ”Towards integrated modelling of urban systems”*, 2014.

[36] P. Bonnel *et al.*, “A survey on the calibration and validation of integrated land use and transportation models,” in *Symposium ”Towards integrated modelling of urban systems”*, Lyon, France: HAL Id: hal-01093254, 2014.

[37] L. Gilquin *et al.*, “Sensitivity Analysis and Optimisation of a Land Use and Transport Integrated Model,” *Journal de la Société Française de Statistique*, 2016.

[38] L. Gilquin *et al.*, “Sensitivity Analysis and Optimisation of a Land Use and Transport Integrated Model To cite this version : Sensitivity Analysis and Optimisation of a Land Use and Transport Integrated Model,” vol. 158, no. 1, pp. 90–110, 2016.

[39] C. Boittin, N. Gaud, V. Hilaire, and D. Meignan, “Particle swarm for calibration of land-use and transport integrated models,” *CUPUM 2015 - 14th International Conference on Computers in Urban Planning and Urban Management*, 2015.

[40] D. B. Lee, “A requiem for large scale modeling,” *Journal Of The American Institute Of Planners*, vol. 39, no. 3. pp. 163–178, 1973. doi: 10.1145/1102945.1102950.

[41] A. Anas and T. Hiramatsu, “The economics of cordon tolling: General equilibrium and welfare analysis,” *Economics of Transportation*, vol. 2, no. 1, pp. 18–37, 2013, doi: 10.1016/j.ecotra.2012.08.002.

[42] M. Batty, “3. Urban Modelling,” 1976.

[43] M. H. Echenique, A. D. J. Flowerdew, J. D. Hunt, T. R. Mayo, I. J. Skidmore, and D. C. Simmonds, “The Meplan models of bilbao, Leeds and Dortmund: Foreign summaries,” *Transp Rev*, vol. 10, no. 4, pp. 309–322, 1990, doi: 10.1080/01441649008716764.

[44] JD Hunt and JE Abraham, “PECAS - for Spatial Economic Modelling Theoretical Formulation,” no. June, 2007.

[45] J. Delons, N. Coulombel, and F. Leurent, “PIRANDELLO an integrated transport and land-use model for the Paris area,” no. July 2008, pp. 1–23, 2008, [Online]. Available: http://halshs.archives-ouvertes.fr/hal-00319087/

[46] D. D. E. L. U. E, T. Capelle, M. Batty, and P. Sturm, “Recherche sur des m ethodes d ’ optimisation pour la mise en place de mod eles int egr de transport et usage des sols Development of optimisation methods for,” 2006.

[47] M. Kryvobokov, A. Mercier, A. Bonnafous, and D. Bouf, “Simulating housing prices with UrbanSim: Predictive capacity and sensitivity analysis,” *Lett Spat Resour Sci*, vol. 6, no. 1, pp. 31–44, 2013, doi: 10.1007/s12076-012-0084-1.

[48] H. Ševčíková, A. E. Raftery, and P. A. Waddell, "Assessing uncertainty in urban simulations using Bayesian melding," *Transportation Research Part B: Methodological*, vol. 41, no. 6, pp. 652–669, 2007, doi: 10.1016/j.trb.2006.11.001.

[49] S. H. Putman, *Policy analysis of transportation and land use.* London: Pion Ltd., 1983.

[50] S. H. Putman, "DRAM Residential Location and Land Use Model: 40 Years ofDevelopment and Application," in *Advances in Spatial Science*, SPRINGER, 2007, pp. 61–76. doi: 10.1007/978-3-642-12788-5_3.

[51] S. Krishnamurthy and K. M. Kockelman, "Propagation of Uncertainty in Transportation Land Use Models Investigation of DRAM-EMPAL and UTPP Predictions in Austin, Texas," *Transp Res Rec*, no. 1831, pp. 219–229, 2003, doi: 10.3141/1831-25.

[52] F. J. Martínez, "MUSSA: Land Use Model for Santiago City," *Transp Res Rec*, vol. 1552, pp. 126–134, 1996.

[53] P. Dutta, M. Saujot, E. Arnaud, B. Lef, and E. Prados, "Uncertainty Propagation and Sensitivity Analysis During Calibration of an Integrated Land Use and Transport Model," *International Journal of Civil and Environmental Engineering*, no. 6, pp. 121–129, 2012.

[54] F. lo FEUDO, "UN SCENARIO TOD POUR LA REGION NORD-PAS-DE- CALAIS. ENSEIGNEMENTS D'UNE MODELISATION INTEGREE TRANSPORT-USAGE DU SOL," 2014. doi: 10.2109/jcersj1950.59.217.

[55] P. Dutta, E. Arnaud, E. Prados, and M. Saujot, "Calibration of an integrated land-use and transportation model using maximum-likelihood estimation," *IEEE Transactions on Computers*, vol. 63, no. 1, pp. 167–178, 2014, doi: 10.1109/TC.2013.168.

[56] T. Capelle, P. Sturm, and A. Vidard, "Formulating LUTI calibration as an optimisation problem: Example of tranus shadow price estimation," *Procedia Eng*, vol. 115, pp. 12–20, 2015, doi: 10.1016/j.proeng.2015.07.349.

[57] T. Capelle, P. Sturm, A. Vidard, and B. Morton, "Contributions to the calibration of integrated land use and transportation models," *CUPUM 2015 - 14th International Conference on Computers in Urban Planning and Urban Management*, 2015.

[58] T. Capelle, "Development of optimization methods for land-use and transportation models," 2018.

[59] L. Feudo, B. Morton, T. Capelle, and E. Prados, "Modelling urban dynamics using luti models: Calibration methodology for the tranus-based model of the grenoble urban region," *Transport and Society - Proceeding of the 22nd International Conference of Hong Kong Society for Transportation Studies, HKSTS 2017*, no. December, pp. 578–586, 2017.

[60] F. lo Feudo, B. Morton, E. Prados, P. Sturm, and T. Capelle, "An operational application of a LUTI model: Scenarization process, implementation and results of a Tranus-based simulation model for the Urban Region of Grenoble," *Transportation Research Procedia*, vol. 41, pp. 178–180, 2019, doi: 10.1016/j.trpro.2019.09.034.

[61] T. Capelle, P. Sturm, A. Vidard, and B. J. Morton, "Calibration of the Tranus land use module: Optimisation-based algorithms, their validation, and parameter selection by statistical model selection," *Comput Environ Urban Syst*, vol. 77, 2019, doi: 10.1016/j.compenvurbsys.2017.04.009.

[62] L. T. Biegler and I. E. Grossmann, "Retrospective on optimization," *Comput Chem Eng*, vol. 28, no. 8, pp. 1169–1192, 2004, doi: 10.1016/j.compchemeng.2003.11.003.

[63] I. E. Grossmann and L. T. Biegler, "Part II. Future perspective on optimization," *Comput Chem Eng*, vol. 28, no. 8, pp. 1193–1218, 2004, doi: 10.1016/j.compchemeng.2003.11.006.

[64] G. Venter, "Review of Optimization Techniques," *Encyclopedia of Aerospace Engineering*, pp. 1–12, 2010, doi: 10.1002/9780470686652.eae495.

[65] J.-L. (1736-1813) Lagrange, *Th´eorie des fonctions analytiques*. Paris: Impr. de la R´epublique, 1797.

[66] S. E. Cox, R. T. Haftka, C. A. Baker, B. Grossman, W. H. Mason, and L. T. Watson, "A Comparison of Global Optimization Methods for the Design of a High-speed Civil Transport," *Journal of Global Optimization*, vol. 21, no. 4, pp. 415–432, 2001, doi: 10.1023/A:1012782825166.

[67] C. A. Floudas and C. E. Gounaris, "A review of recent advances in global optimization," *Journal of Global Optimization*, vol. 45, no. 1, pp. 3–38, 2009, doi: 10.1007/s10898-008-9332-8.

[68] A. Neumaier, "Complete search in continuous global optimization and constraint satisfaction," *Acta Numerica*, vol. 13, pp. 271–369, 2004, doi: 10.1017/S0962492904000194.

[69] K. V. P. · R. M. S. J. A. Lampinen, *Differential Evolution APractical Approach to Global Optimization*, vol. 12, no. 2. Springer, 2004. doi: 10.1162/evco.2004.12.2.269.

[70] B. Hartke, "Global optimization," *Wiley Interdiscip Rev Comput Mol Sci*, vol. 1, no. 6, pp. 879–887, 2011, doi: 10.1002/wcms.70.

[71] C. A. C. Coello, G. B. Lamont, and D. A. van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*. 2007. doi: 10.1007/978-0-387-36797-2.

[72] Zbigniew Michalewics, *Genetic Algorithms + Data Structures = Evolution Programs*, Third edit. 1996.

[73] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[74] T. Bäck, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. USA: Oxford University Press, Inc., 1996.

[75] S. Luke, "Issues in Scaling Genetic Programming: Breeding Strategies, Tree Generation, and Code Bloat," University of Maryland, 2000. [Online]. Available: http://www.cs.gmu.edu/~sean/papers/thesis2p.pdf

[76] R. Kicinger, T. Arciszewski, and K. de Jong, "Evolutionary computation and structural design: A survey of the state-of-the-art," *Comput Struct*, vol. 83, no. 23–24, pp. 1943–1978, 2005, doi: 10.1016/j.compstruc.2005.03.002.

[77] J. H. Holland, *Adaptation in Natural and Artificial Systems_ An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The MIT Press, 1992. [Online]. Available: https://books.google.com.br/books?id=JE5RAAAAMAAJ

[78] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, pp. 1942–1948 vol.4, 1995, doi: 10.1109/ICNN.1995.488968.

[79] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997, doi: 10.1023/A:1008202821328.

[80] C. Darwin, "The origin of species by means of natural selection." London, p. 556, 1859.

[81] A. K. Palit and D. Popović, *Computational intelligence in time series forecasting : theory and engineering applications*. Springer, 2005.

[82] T. Weise, *Global Optimization Algorithms-Theory and Application-2 n d d E d*. 2009. [Online]. Available: http://www.it-weise.de/

[83] I. Strnad and M. Žura, "Genetic algorithms application to EVA mode choice model parameters estimation," vol. 5, no. 3, pp. 533–541, 2011.

[84] N. Dadashzadeh, M. Ergun, S. Kesten, and M. Žura, "An automatic calibration procedure of driving behavior parameters in the presence of high bus demand," *Promet - Traffic & Transportation*, vol. 31, no. 5, pp. 491–502, 2019, doi: 10.7307/ptt.v31i5.3100.

[85] N. Dadashzadeh, M. Ergun, A. S. Kesten, and M. Žura, "Improving the calibration time of traffic simulation models using parallel computing technique," in *MT-ITS2019 6th International Conference on Models and Technologies for Intelligent Transportation Systems*, R. Kucharski and A. Szarata, Eds., Kraków, Poland: IEEE Xplore, 2019, p. 54.

[86] X. Ding, M. Zheng, and X. Zheng, "The application of genetic algorithm in land use optimization research: A review," *Land (Basel)*, vol. 10, no. 5, pp. 1–21, 2021, doi: 10.3390/land10050526.

[87] M. M. Millonas, "Swarms, Phase Transitions, and Collective Intelligence," *arXiv: Adaptation and Self-Organizing Systems*, 1993.

[88] K. E. Parsopoulos and M. N. Vrahatis, *Particle Swarm Optimization and Intelligence*, vol. 270. 2010. [Online]. Available: http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-61520-666-7

[89] E. Elbeltagi, T. Hegazy, and D. Grierson, "Comparison among five evolutionary-based optimization algorithms," *Advanced Engineering Informatics*, vol. 19, no. 1, pp. 43–53, 2005, doi: 10.1016/j.aei.2005.01.004.

[90] S. Palmer, "Evolutionary Algorithms and Computational Methods for Derivatives Pricing," *Doctoral thesis, UCL (University College London).* , 2019.

[91] A. Ali, S. Siddharth, Z. Syed, and N. El-Sheimy, "Swarm optimization-based magnetometer calibration for personal handheld devices," *Sensors (Switzerland)*, vol. 12, no. 9, pp. 12455–12472, 2012, doi: 10.3390/s120912455.

[92] C. Blum and D. Merkle(Eds.), *Swarm Intelligence Introduction and Applications*, vol. 5, no. 10. Berlin, Germany: Springer, 2008.

[93] R. Havangi, M. A. Nekoui, and M. Teshnehlab, "A Multi Swarm Particle Filter for Mobile Robot Localization," *International Journal of Computer Science Issues*, vol. 7, no. 3, pp. 15–22, 2010, [Online]. Available: www.IJCSI.org

[94] M. A. Asl, "Active Contour Optimization Using Particle Swarm Optimizer," pp. 522–523, 2006.

[95] L. A. Maceachern and T. Manku, "Genetic algorithms for active contour optimization," pp. 11–14, 1998.

[96] G.-R. SE. Cruz-Aceves I, Aviña-Cervantes JG, López-Hernández JM, "Multiple Active Contours Driven by Particle Swarm Optimization for Cardiac Medical Image Segmentation," *Comput Math Methods Med*, vol. 2013, p. 13, 2013, [Online]. Available: http://dx.doi.org/10.1155/2013/132953%0D

[97] C. Tseng, J. Hsieh, and J. Jeng, "Expert Systems with Applications Active contour model via multi-population particle swarm optimization," *Expert Syst Appl*, vol. 36, no. 3, pp. 5348–5352, 2009, doi: 10.1016/j.eswa.2008.06.114.

[98] J. Liang and P. N. Suganthan, "Dynamic Multi-Swarm Particle Swarm Optimizer with a Novel Constraint- Dynamic Multi-Swarm Particle Swarm Optimizer with a Novel Constraint-Handling Mechanism," no. May 2014, 2019, doi: 10.1109/CEC.2006.1688284.

[99] R. A. Krohling and S. Coelho, "Coevolutionary Particle Swarm Optimization Using Gaussian Distribution for Solving Constrained Optimization Problems," vol. 36, no. 6, pp. 1407–1416, 2006.

[100] D. Sedighizadeh and E. Masehian, "6..Pso Methods, Taxonomy and Applications (1)," vol. 1, no. 5, pp. 486–502, 2009.

[101] A. Gopal, M. M. Sultani, and J. C. Bansal, "On Stability Analysis of Particle Swarm Optimization Algorithm," *Arab J Sci Eng*, vol. 45, no. 4, pp. 2385–2394, 2020, doi: 10.1007/s13369-019-03991-8.

[102] D. P. Rini and S. M. Shamsuddin, "Particle Swarm Optimization: Technique, System and Challenges," *Int J Appl Inf Syst*, vol. 1, no. 1, pp. 33–45, 2011, doi: 10.5120/ijais-3651.

[103] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006, doi: 10.1109/TEVC.2005.857610.

[104] A. Slowik, "Application of an Adaptive Differential Evolution Algorithm With Multiple Trial Vectors to Artificial Neural Network Training," vol. 58, no. 8, pp. 3160–3167, 2011.

[105] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, Dec. 2006, doi: 10.1109/TEVC.2006.872133.

[106] M. M. Ali, C. Khompatraporn, and Z. B. Zabinsky, "A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems," in *Journal of Global Optimization*, Apr. 2005, pp. 635–672. doi: 10.1007/s10898-004-9972-2.

[107] K. v. Price, R. M. Storn, and J. A. Lampinen, *Differential evolution : a practical approach to global optimization*. Springer, 2005.

[108] A. P. Engelbrecht, *Computational Intelligence:An Introduction*, Second Edi. England: Wiley, 2007.

[109] M. Iwan, R. Akmeliawati, T. Faisal, and H. M. A. A. Al-Assadi, "Performance comparison of differential evolution and particle swarm optimization in constrained optimization," *Procedia Eng*, vol. 41, no. Iris, pp. 1323–1328, 2012, doi: 10.1016/j.proeng.2012.07.317.

[110] R. Ugolotti *et al.*, "Particle Swarm Optimization and Differential Evolution for model-based object detection To cite this version : HAL Id : hal-01221292 Particle Swarm Optimization and Differential Evolution for Model-based Object Detection," 2015.

[111] V. Kachitvichyanukul, "GA/PSO/DE..Comparison of Three Evolutionary Algorithms: GA, PSO, and DE," *Industrial Engineering and Management Systems*, vol. 11, no. 3, pp. 215–223, 2012, doi: 10.7232/iems.2012.11.3.215.

[112] A. Al-Dujaili, M. R. Tanweer, and S. Suresh, "DE vs. PSO: A performance assessment for expensive problems," *Proceedings - 2015 IEEE Symposium Series on Computational Intelligence, SSCI 2015*, pp. 1711–1718, 2015, doi: 10.1109/SSCI.2015.240.

[113] K. N. Porfyri, I. K. Nikolos, A. I. Delis, and M. Papageorgiou, "Calibration of a second-order traffic flow model using a metamodel-assisted Differential Evolution algorithm," *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 366–371, 2016, doi: 10.1109/ITSC.2016.7795581.

[114] Y. Li, S. Wang, and B. Yang, "An improved differential evolution algorithm with dual mutation strategies collaboration," *Expert Syst Appl*, vol. 153, 2020, doi: 10.1016/j.eswa.2020.113451.

[115] W. Gong, Z. Cai, and D. Liang, "Adaptive Ranking Mutation Operator Based Differential Evolution for Constrained Optimization," *IEEE Trans Cybern*, vol. 45, no. 4, pp. 716–727, Apr. 2015, doi: 10.1109/TCYB.2014.2334692.

[116] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," in *Applied Soft Computing Journal*, Mar. 2011, pp. 1679–1696. doi: 10.1016/j.asoc.2010.04.024.

[117] K. Opara and J. Arabas, "Comparison of mutation strategies in Differential Evolution – A probabilistic perspective," *Swarm Evol Comput*, vol. 39, pp. 53–69, Apr. 2018, doi: 10.1016/j.swevo.2017.12.007.

[118] S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 482–500, Apr. 2012, doi: 10.1109/TSMCB.2011.2167966.

[119] Thomas Bäck, *Evolutionary algorithms in theory and practice - evolution strategies, evolutionary programming, genetic algorithms*, 1st ed. New Your, Oxford: Oxford University Press, 1996.

[120] S. L. Wang, T. F. Ng, N. A. Jamil, S. M. Samuri, R. Mailok, and B. Rahmatullah, "Self-adapting approach in parameter tuning for differential evolution," in *TAAI 2015 - 2015 Conference on Technologies and Applications of Artificial Intelligence*, Institute of Electrical and Electronics Engineers Inc., Feb. 2016, pp. 113–119. doi: 10.1109/TAAI.2015.7407109.

[121] J. Aalto and J. Lampinen, "A population adaptation mechanism for differential evolution algorithm," in *Proceedings - 2015 IEEE Symposium Series on Computational Intelligence, SSCI 2015*, Institute of Electrical and Electronics Engineers Inc., 2015, pp. 1514–1521. doi: 10.1109/SSCI.2015.214.

[122] J. Zhang and A. C. (Arthur C.) Sanderson, *Adaptive differential evolution : a robust approach to multimodal problem optimization*. Springer, 2009.

[123] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009, doi: 10.1109/TEVC.2008.927706.

[124] A. S. H. W. F. L. E.-P. L. Ying Liu, *Advances of ComputationalIntelligence in Industrial Systems*. Verlag Berlin Heidelberg: Springer, 2008.

[125] Vitaliy. Feoktistov, *Differential evolution : in search of solutions*. Springer Science+Business Media, 2006.

[126] M. N. Katehakis and A. F. Veinott, "MULTI-ARMED BANDIT PROBLEM: DECOMPOSITION AND COMPUTATION.," *Mathematics of Operations Research*, vol. 12, no. 2, pp. 262–268, 1987, doi: 10.1287/moor.12.2.262.

[127] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential Model-Based Optimization for General Algorithm Configuration."

[128] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differential evolution-An updated survey," *Swarm Evol Comput*, vol. 27, pp. 1–30, Apr. 2016, doi: 10.1016/j.swevo.2016.01.004.

[129] M. Yang, C. Li, Z. Cai, and J. Guan, "Differential evolution with auto-enhanced population diversity," *IEEE Trans Cybern*, vol. 45, no. 2, pp. 302–315, Feb. 2015, doi: 10.1109/TCYB.2014.2339495.

[130] Y. L. Li, Z. H. Zhan, Y. J. Gong, W. N. Chen, J. Zhang, and Y. Li, "Differential Evolution with an Evolution Path: A DEEP Evolutionary Algorithm," *IEEE Trans Cybern*, vol. 45, no. 9, pp. 1798–1810, Sep. 2015, doi: 10.1109/TCYB.2014.2360752.

[131] S. M. Guo, C. C. Yang, P. H. Hsu, and J. S. H. Tsai, "Improving differential evolution with a successful-parent-selecting framework," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 717–730, Oct. 2015, doi: 10.1109/TEVC.2014.2375933.

[132] E. Mezura-Montes, M. Reyes-Sierra, and C. A. Coello, "Multi-objective optimization using differential evolution: A survey of the state-of-the-art," *Studies in Computational Intelligence*, vol. 143, no. June 2014, pp. 173–196, 2008, doi: 10.1007/978-3-540-68830-3_7.

[133] J. Nossent, P. Elsen, and W. Bauwens, "Environmental Modelling & Software Sobol ' sensitivity analysis of a complex environmental model," *Environmental Modelling and Software*, vol. 26, no. 12, pp. 1515–1525, 2011, doi: 10.1016/j.envsoft.2011.08.010.

[134] M. P. Butler, P. M. Reed, K. Fisher-vanden, K. Keller, and T. Wagener, "Environmental Modelling & Software Identifying parametric controls and dependencies in integrated assessment models using global sensitivity analysis," *Environmental Modelling and Software*, vol. 59, pp. 10–29, 2014, doi: 10.1016/j.envsoft.2014.05.001.

[135] N. A. S. Hamm, J. W. Hall, and M. G. Anderson, "Variance-based sensitivity analysis of the probability of hydrologically induced slope instability," vol. 32, pp. 803–817, 2006, doi: 10.1016/j.cageo.2005.10.007.

[136] C. D. a R. Pastres a'l, K. Chart b, C. Solidoro c, "Global sensitivity analysis of a shallow-water 3D eutrophication model," vol. 117, pp. 62–74, 1999.

[137] T. G. Nguyen and J. L. de Kok, "Systematic testing of an integrated systems model for coastal zone management using sensitivity and uncertainty analyses," vol. 22, pp. 1572–1587, 2007, doi: 10.1016/j.envsoft.2006.08.008.

[138] P. Dutta, M. Saujot, E. Arnaud, B. Lef, and E. Prados, "Uncertainty Propagation and Sensitivity Analysis During Calibration of an Integrated Land Use and Transport Model," *International Journal of Civil and Environmental Engineering*, vol. 6, no. 6, pp. 121–129, 2012.

[139] I. M. Sobol, "Sensitivity analysis for nonlinear mathematical models," *Math. Model. Computer.Exp*, vol. 1, no. 4, pp. 407–414, 1993, doi: 10.18287/0134-2452-2015-39-4-459-461.

[140] A. Saltelli, "Making best use of model evaluations to compute sensitivity indices," *Comput Phys Commun*, vol. 145, no. 2, pp. 280–297, 2002, doi: 10.1016/S0010-4655(02)00280-1.

[141] J. Herman and W. Usher, "SALib: An open-source Python library for Sensitivity Analysis," *The Journal of Open Source Software*, vol. 2, no. 9, p. 97, 2017, doi: 10.21105/joss.00097.

[142] S. McKeen *et al.*, "Assessment of an ensemble of seven real-time ozone forecasts over eastern North America during the summer of 2004," *Journal of Geophysical Research Atmospheres*, vol. 110, no. 21, pp. 1–16, 2005, doi: 10.1029/2005JD005858.

[143] N. H. Savage *et al.*, "Air quality modelling using the Met Office Unified Model (AQUM OS24-26): Model description and initial evaluation," *Geosci Model Dev*, vol. 6, no. 2, pp. 353–372, 2013, doi: 10.5194/gmd-6-353-2013.

[144] A. Chatterjee, R. J. Engelen, S. R. Kawa, C. Sweeney, and A. M. Michalak, "Background error covariance estimation for atmospheric CO2 data assimilation," *Journal of Geophysical Research Atmospheres*, vol. 118, no. 17, pp. 10,140-10,154, 2013, doi: 10.1002/jgrd.50654.

[145] J. Hourdakis, P. G. Michalopoulos, and J. Kottommannil, "A PRACTICAL PROCEDURE FOR CALIBRATING MICROSCOPIC By," *Transportation Research Board*, vol. 1852, no. January, pp. 130–139, 2003.

[146] J. Ma, H. Dong, and H. M. Zhang, "Calibration of microsimulation with heuristic optimization methods," *Transp Res Rec*, no. 1999, pp. 208–217, 2007, doi: 10.3141/1999-22.

[147] X. Lu, J. Lee, D. Chen, J. Bared, D. Dailey, and S. E. Shladover, "Freeway Micro-simulation Calibration : Case Study Using Aimsun and VISSIM with Detailed Field Data," *Transportation Research Board 93rd Annual Meeting. January 12-16, Washington, D.C.*, no. January, pp. 1–17, 2014.

[148] "Freeway Micro-simulation Calibration Case Study Using VISSIM.pdf."

[149] B. Ciuffo, V. Punzo, and V. Torrieri, "Comparison of simulation-based and model-based calibrations of traffic-flow microsimulation models," *Transp Res Rec*, no. 2088, pp. 36–44, 2008, doi: 10.3141/2088-05.

[150] Y. Hollander and R. Liu, "The principles of calibrating traffic microsimulation models," *Transportation (Amst)*, vol. 35, no. 3, pp. 347–362, 2008, doi: 10.1007/s11116-007-9156-2.

[151] J. B. Lee and K. Ozbay, "New calibration methodology for microscopic traffic simulation using enhanced simultaneous perturbation stochastic approximation approach," *Transp Res Rec*, no. 2124, pp. 233–240, 2009, doi: 10.3141/2124-23.

[152] M. Yu and W. (David) Fan, "Calibration of microscopic traffic simulation models using metaheuristic algorithms," *International Journal of Transportation Science and Technology*, vol. 6, no. 1, pp. 63–77, 2017, doi: 10.1016/j.ijtst.2017.05.001.

[153] I. Strnad and M. Žura, "Genetic algorithms application to EVA mode choice model parameters estimation," vol. 5, no. 3, pp. 533–541, 2011.

[154] L. de Jonge, "Calibrating parameters in the MISCAN model using a genetic algorithm," 2019.

[155] H. Pang *et al.*, "Calibration of three-axis magnetometers with differential evolution algorithm," *J Magn Magn Mater*, vol. 346, pp. 5–10, 2013, doi: 10.1016/j.jmmm.2013.06.051.

[156] jiqiang, "A Unified Differential Evolution Algorithm for Global Optimization," no. 7, pp. 1–8, 2014, [Online]. Available: papers3://publication/uuid/F77AA14A-7CA6-473C-92A8-E45A038B8169

[157] A. D. Ukhov, "Differential Evolution : A Tool for Global Optimization," vol. 16, pp. 3–44, 2016.

[158] P. Rodriguez-Mier, "A tutorial on Differential Evolution with Python," *GitHub*, pp. 1–18, 2017.

[159] P. Kaelo and M. M. Ali, "Differential evolution algorithms using hybrid mutation," *Comput Optim Appl*, vol. 37, no. 2, pp. 231–246, Jun. 2007, doi: 10.1007/s10589-007-9014-3.

[160] R. Gaemperle, S. D. Mueller, and P. Koumoutsakos, "A Parameter Study for Differential Evolution," in *Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*, WSEAS, 2002, p. 293298. [Online]. Available: http://www.cse-lab.ethz.ch/wp-content/papercite-data/pdf/gaemperle2002a.pdf

[161] M. K. Heris, "Yarpiz Evolutionary Algorithms," *YPEA*, 2019. https://yarpiz.com/477/ypea-yarpiz-evolutionary-algorithms), Yarpiz, 2019

[162] A. Hassanat, K. Almohammadi, E. Alkafaween, E. Abunawas, A. Hammouri, and V. B. S. Prasath, "Choosing mutation and crossover ratios for genetic algorithms-a review with a new dynamic approach," *Information (Switzerland)*, vol. 10, no. 12, 2019, doi: 10.3390/info10120390.

[163] W. Teekeng and P. Unkaw, "A new hybrid model of PSO and de algorithm for data classification," *Proceedings - 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2017*, pp. 47–51, 2017, doi: 10.1109/SNPD.2017.8022699.

[164] W. Y. Lin, "A GA-DE hybrid evolutionary algorithm for path synthesis of four-bar linkage," *Mech Mach Theory*, vol. 45, no. 8, pp. 1096–1107, 2010, doi: 10.1016/j.mechmachtheory.2010.03.011.

[165] B. Xin, J. Chen, J. Zhang, H. Fang, and Z. H. Peng, "Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: A review and taxonomy," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 42, no. 5, pp. 744–767, 2012, doi: 10.1109/TSMCC.2011.2160941.

[166] W. J. Zhang and X. F. Xie, "DEPSO: Hybrid particle swarm with differential evolution operator," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, no. 1, pp. 3816–3821, 2003, doi: 10.1109/icsmc.2003.1244483.

[167] L. Xiao and X. Zuo, "Multi-DEPSO: A de and PSO based hybrid algorithm in dynamic environments," *2012 IEEE Congress on Evolutionary Computation, CEC 2012*, pp. 10–15, 2012, doi: 10.1109/CEC.2012.6256178.

[168] F. Hameed, S. Mariyam, and H. Shamsuddin, "A New Hybrid Particle Swarm Optimization-Differential Evolution Algorithm," *Life Sci J*, vol. 18, no. 6, pp. 1097–8135, 2021, doi: 10.7537/marslsj180621.03.