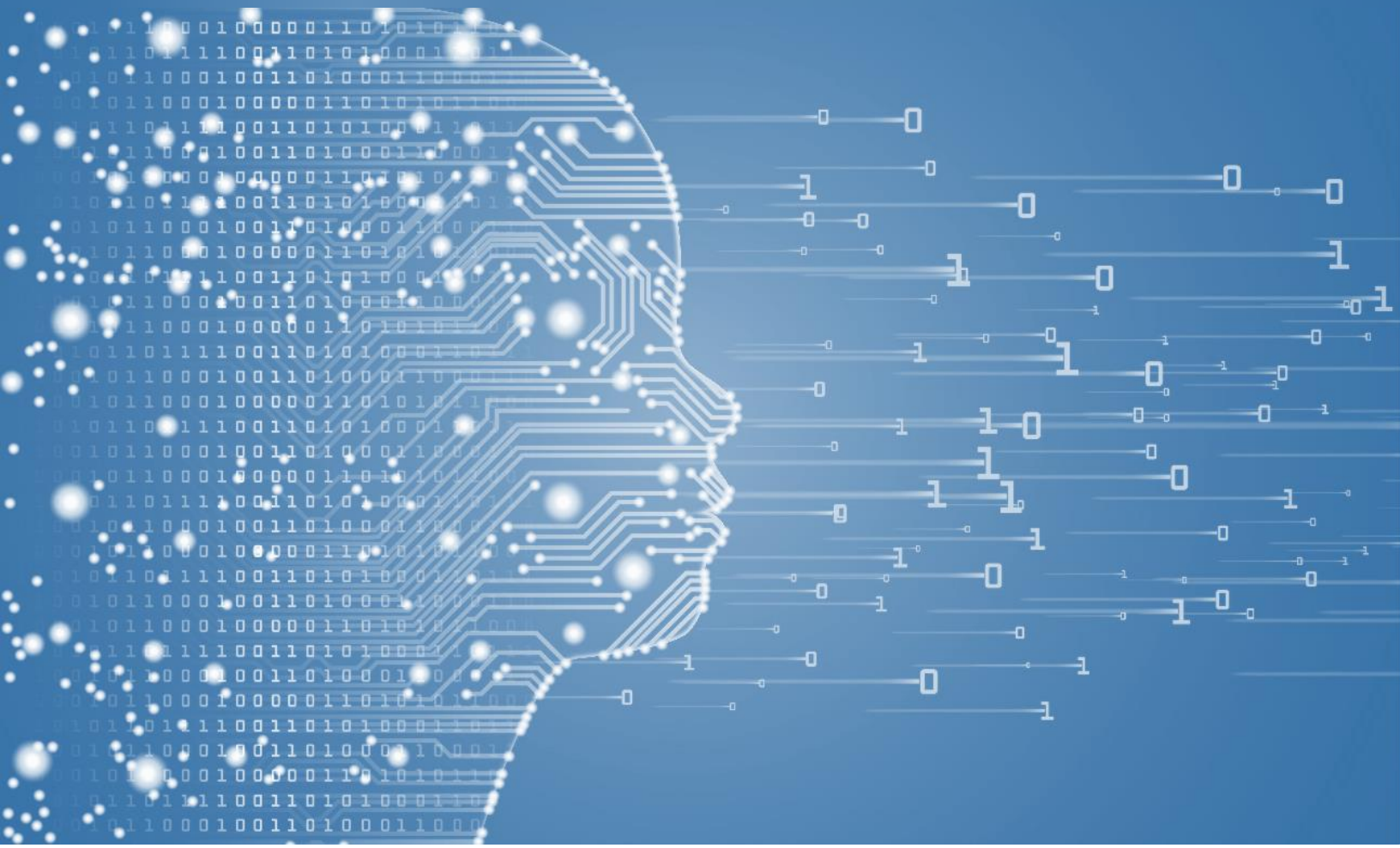
CrORR

Croatian Operational Research Review

vol. 15 / no. 2 / 2024

ISSN (print): 1848 – 0225

ISSN (online): 1848 – 9931



CRORS

CROATIAN OPERATIONAL RESEARCH SOCIETY

ISSN: 1848–0225 Print
ISSN: 1848–9931 Online
UDC: 519.8 (063)
Abbreviation: Croat. Oper. Res. Rev.

Publisher:
CROATIAN OPERATIONAL RESEARCH SOCIETY

Co-publishers:
University of Zagreb, Faculty of Economics & Business
J. J. Strossmayer University of Osijek, Faculty of Economics in Osijek
University of Split, Faculty of Economics, Business and Tourism
J. J. Strossmayer University of Osijek, School of Applied Mathematics and Informatics
University of Zagreb, Faculty of Organization and Informatics

Croatian Operational Research Review

Volume 15 (2024)
Number 2

Zagreb, 2024

Croatian Operational Research Review is viewed/indexed in: Current Index to Statistics, DOAJ, EBSCO host, EconLit, Genamics Journal Seek, INSPEC, Mathematical Reviews, Current Mathematical Publications (MathSciNet), Proquest, SCOPUS, Web of Science Emerging Sources Citation Index (WoS ESCI), Zentralblatt für Mathematik/Mathematics Abstracts (CompactMath).

ISSN: 1848–0225 Print
ISSN: 1848–9931 Online
UDC: 519.8 (063)
Abbreviation: Croat. Oper. Res. Rev.

Croatian Operational Research Review

Croatian Operational Research Review (CRORR) is an open access scientific journal published bi-annually. Starting from 2014, Number 1, the main publisher of the CRORR journal is the Croatian Operational Research Society (CRORS). Co-publishers of the journal are:

- ◇ University of Zagreb, Faculty of Economics & Business,
- ◇ J. J. Strossmayer University of Osijek, Faculty of Economics in Osijek,
- ◇ University of Split, Faculty of Economics, Business and Tourism,
- ◇ J. J. Strossmayer University of Osijek, School of Applied Mathematics and Informatics,
- ◇ University of Zagreb, Faculty of Organization and Informatics.

Aims and Scope of the Journal

The aim of the Croatian Operational Research Review journal is to provide high quality scientific papers covering the theory and application of operational research and related areas, mainly quantitative methods and machine learning. The scope of the journal is focused, but not limited to the following areas: combinatorial and discrete optimization, integer programming, linear and nonlinear programming, multiobjective and multicriteria programming, statistics and econometrics, macroeconomics, economic theory, games control theory, stochastic models and optimization, banking, finance, insurance, simulations, information and decision support systems, data envelopment analysis, neural networks and fuzzy systems, and practical OR and application.
Occasionally, special issues are published dedicated to a particular area of OR or containing selected papers from the International Conference on Operational Research.
Papers blindly reviewed and accepted by two independent reviewers are published in this journal.

Publication Ethics and Publication Malpractice StatementPublication Ethics and Publication Malpractice Statement

CRORR journal is committed to ensuring ethics in publications and quality of articles. Conformance to standards of ethical behaviour is therefore expected of all parties involved: Authors, Editors, Reviewers, and the Publisher. Before submitting or reviewing a paper read our Publication ethics and Publication Malpractice Statement available at the journal homepage: http://www.hdoi.hr/crorr-journal.
The journal is financially supported by the Ministry of Science, Education and Sports of the Republic of Croatia, and by its co-publishers. All data referring to the journal with full papers texts since the beginning of its publication in 2010 can be found in the Hrcak database – Portal of scientific journals of Croatia (see http://hrcak.srce.hr./crorr).

Open Access Statement

Croatian Operational Research Review is an open access journal which means that all content is freely available without charge to the user or his/her institution. Users are allowed to read, download, copy, distribute, print, search, or link to the full texts of the articles in the journal without asking prior permission from the publisher or the author. This is in accordance with the BOAI definition of open access. The journal has an open access to full text of all papers through our Open Journal System (http://hrcak.srce.hr/ojs/index.php/crorr) and the Hrcak database (http://hrcak.srce.hr/crorr).

Croatian Operational Research Review is viewed/indexed by:

- ◇ Current Index to Statistics
- ◇ Directory of Open Access Journals (DOAJ)
- ◇ EBSCO host
- ◇ EconLit
- ◇ Genamics Journal Seek
- ◇ INSPEC
- ◇ Mathematical Reviews, Current Mathematical Publications (MathSciNet)
- ◇ Proquest
- ◇ SCOPUS
- ◇ Web of Science Emerging Sources Citation Index (WoS ESCI)
- ◇ Zentralblatt für Mathematik/Mathematics Abstracts (CompactMath)

Editorial Office and Address:

Editors-in-Chief:	Blanka Škrabić Perić Tea Šestanović
Co-Editors:	Josip Arnerić Anita Čeh Časni Mirjana Čižmešija Mateja Đumić Danijel Grahovac Goran Lešaja Ivan Papić Snježana Pivac
Technical Editors:	Ivana Mravak Tea Kalinić Milićević Marija Vuković
Design:	Tea Kalinić Milićević
Cover image:	www.123rf.com

Contact:
Croatian Operational Research Society (CRORS), Trg J. F. Kennedyja 6,
HR-10000 Zagreb, Croatia
URL: http://www.hdoi.hr/crorr-journal
e-mail: crorr.journal@gmail.com

Editorial Board

Violeta Cvetkoska, Faculty of Economics, Ss. Cyril and Methodius, University in Skopje, Republic of North Macedonia, e-mail: vcvetkoska@eccf.ukim.edu.mk
Maja Ćukušić, Faculty of Economics, Business and Tourism, University of Split, Croatia, e-mail: maja.cukusic@efst.hr
Anita Čeh Časni, Faculty of Economics & Business, University of Zagreb, Croatia, e-mail: aceh@efzg.hr
Mirjana Čižmešija, Faculty of Economics & Business, University of Zagreb, Croatia, e-mail: mcizmesija@efzg.hr
Ksenija Dumičić, Faculty of Economics & Business, University of Zagreb, Croatia, e-mail: kdumicic@efzg.hr
Margareta Gardijan Kedžo, Faculty of Economics & Business, University of Zagreb, Croatia, e-mail: mgardijan@efzg.hr
Danijel Grahovac, School of Applied Mathematics and Informatics, J. J. Strossmayer University of Osijek, Croatia, e-mail: dgrahova@mathos.hr
Tihomir Hunjak, Faculty of Organization and Informatics, University of Zagreb, Croatia, e-mail: tihomir.hunjak@foi.hr
Mario Jadrić, Faculty of Economics, Business and Tourism, University of Split, Croatia, e-mail: jadric@efst.hr
Nikola Kadoić, Faculty of Organization and Informatics, University of Zagreb, Croatia, e-mail: nikola.kadoic@foi.unizg.hr
Vedran Kojić, Faculty of Economics & Business, University of Zagreb, Croatia, e-mail: vkojic@efzg.hr
Ulrike Leopold-Wildburger, Institut für Statistik un Operations Research, Karl-Franzens-Universität Graz, Austria, e-mail: ulrike.leopold@uni-graz.at
Goran Lešaja, Department of Mathematical Sciences, Georgia Southern University, USA, e-mail: goran@georgiasouthern.edu
Zrinka Lukač, Faculty of Economics & Business, University of Zagreb, Croatia, e-mail: zlukac@efzg.hr
Robert Manger, Department of Mathematics, University of Zagreb, Croatia, e-mail: manger@math.hr
Josip Mesarić, Faculty of Economics in Osijek, J. J. Strossmayer University of Osijek, Croatia, e-mail: mesaric@efos.hr
Tea Mijač, Faculty of Economics, Business and Tourism, University of Split, Croatia, e-mail: maja.cukusic@efst.hr
Luka Neralić, Faculty of Economics & Business, University of Zagreb, Croatia, e-mail: lneralic@efzg.hr
Snježana Pivac, Faculty of Economics, Business and Tourism, University of Split, Croatia, e-mail: spivac@efst.hr
Krunoslav Puljić, Faculty of Economics & Business, University of Zagreb, Croatia, e-mail: kpuljic@efzg.hr
Jose Rui Figueria, Instituto Superior Tecnico, Technical University of Lisbon, Portugal, e-mail: figueira@ist.utl.pt
Kristian Sabo, School of Applied Mathematics and Informatics, J. J. Strossmayer University of Osijek, Croatia, e-mail: ksabo@mathos.hr
Rudolf Scitovski, School of Applied Mathematics and Informatics, J. J. Strossmayer University of Osijek, Croatia, e-mail: scitowsk@mathos.hr
Darko Skorin-Kapov, School of Business, Adelphi University, Garden City, USA, e-mail: skorin@adelphi.edu
Jadranka Skorin-Kapov, W. A. Harriman School for Management and Policy, State University of New York at Stony Brook, USA, e-mail: jskorin@notes.cc.sunysb.edu
Petar Sorić, Faculty of Economics & Business, University of Zagreb, Croatia, e-mail:

psoric@efzg.hr
Greys Sošić, Information and Operations Management Department, The Marshall School of Business, University of Southern California, Los Angeles, CA, USA, e-mail: sosic@marshall.usc.edu
Kenneth Sörensen, Department of Engineering Management, University of Antwerp, Belgium, e-mail: kenneth.sorensen@uantwerpen.be
Alemka Šegota, Faculty of Economics, University of Rijeka, Croatia, e-mail: alemka.segota@efri.hr
Tea Šestanović, University of Split, Faculty of Economics, Business and Tourism, Croatia, e-mail: tea.sestanovic@efst.hr
Blanka Škrabić Perić, Faculty of Economics, Business and Tourism, University of Split, Croatia, e-mail: blanka.skrabic@efst.hr
Kristina Šorić, Rochester Institute of Technology Croatia, Croatia, e-mail: ksoric@zsem.hr
Richard Wendell, Joseph M. Katz Graduate School of Business, University of Pittsburgh, USA, e-mail: wendell@katz.pitt.edu
Lidija Zadnik Stirn, Biotechnical Faculty, University of Ljubljana, Slovenia, e-mail: Lidija.Zadnik@bf.uni-lj.si
Sanjo Zlobec, Department of Mathematics and Statistics, McGill University, Canada, e-mail: zlobec@math.mcgill.ca
Nikolina Žajdela Hrustek, University of Zagreb, Faculty of Organization and Informatics, Croatia, e-mail: nikolina.zajdela@foi.unizg.hr
Berislav Žmuk, Faculty of Economics & Business, University of Zagreb, Croatia, e-mail: bzmuk@efzg.hr

Croatian Operational Research Review

Volume 15 (2024), Number 2

CONTENTS

A fresh look at a randomized massively parallel graph coloring algorithm

Boštjan Gabrovšek[1,2] **and Janez Žerovnik**[1,3,*]

[1] *University of Ljubljana, Faculty of Mechanical Engineering, Aškerčeva 6, 1000 Ljubljana, Slovenia*

[2] *Institute of Mathematics, Physics and Mechanics, Jadranska ulica 19, 1000 Ljubljana, Slovenia*
E-mail: ⟨bostjan.gabrovsek@fs.uni-lj.si⟩

[3] *Rudolfovo – Science and Technology Centre Novo mesto, Podbreznik 15, 8000 Novo mesto, Slovenia*
E-mail: ⟨janez.zerovnik@fs.uni-lj.si⟩

Abstract. Petford and Welsh introduced a sequential heuristic algorithm to provide an approximate solution to the NP-hard graph coloring problem. The algorithm is based on the antivoter model and mimics the behavior of a physical process based on a multi-particle system of statistical mechanics. It was later shown that the algorithm can be implemented in a massively parallel model of computation. The increase in computational processing power in recent years allows us to perform an extensive analysis of the algorithms on a larger scale, leading to the possibility of a more comprehensive understanding of the behavior of the algorithm, including the phase transition phenomena.

Keywords: combinatorial optimization, graph coloring, randomized local search procedure, temperature

Received: January 25, 2024; accepted: June 4, 2024; available online: October 7, 2024

DOI: 10.17535/crorr.2024.0009

1. Introduction

Graph coloring is one of the most studied NP-hard problems in combinatorial optimization [17]. In addition to its purely combinatorial appeal, the graph coloring problem is widely used in many engineering projects, including various practical problems such as planning and scheduling, timetabling, and frequency assignment [1, 18]. It is generally believed that NP-hard problems cannot be solved optimally in times that are polynomially bounded functions of the input size. However, the conjecture P≠NP? is still an open problem. This conjecture is one of the most important problems in contemporary mathematics because it is on the list of seven millennium problems for which a prize of 1 million dollars is offered by the Clay Institute [9]. NP-complete problems are decision problems with the following property: if any of them enjoys a polynomial time algorithm giving the exact solution, then P=NP, thus solving the P versus NP problem [15, 12]. On the other hand, it is widely believed that P≠NP, in other words, that there is no polynomial time algorithm for any NP-complete problem. It also holds true for NP-hard problems, i.e., problems that are at least as difficult as an NP-complete problem. Therefore, there is great interest in heuristic algorithms that can find near-optimal solutions to the graph coloring problem (or any other NP-hard problem) within reasonable running times for a large number of instances [11].

Threshold phenomena have attracted a lot of attention in the context of random combinatorial problems [8] and in theoretical physics [21]. In statistical physics, phase transitions have

*Corresponding author.

been studied for more than a century. Let us only mention here that spin glasses, a purely theoretical concept, have triggered a new branch of theoretical physics that resulted in a Nobel Prize for Giorgio Parisi in 2022 [19]. The Ising model is a statistical model that can be used to describe the energy of a system of atoms arranged in a lattice, where the interaction between these atoms is governed by the interaction between their spins and a possible external magnetic field. This model and some more complicated versions of it can then be used to describe the behavior of spin glasses. It is well-known that the Ising model is in general NP hard [3]. On the other hand, the Ising model is closely related to graph theory, in particular to the graph coloring problem [3]. Graph coloring with $k = 3$ colors has been considered in several papers, see [5, 2] and references therein. In a study of random graphs it was found that the phase transition is closely related to the mean degree of the graphs. In the same paper, the analysis implies that the hardest instances are among random graphs with an average degree between 4.42 and $\alpha_{crit} \approx 4.7$.

Based on the antivoter model of Donnely and Welsh [6], a randomized algorithm for graph coloring that performs very well on some random graphs [20, 27] was proposed by Petford and Welsh. The algorithm was later successfully tested on some other types of graphs [23]. With some straightforward modifications, the algorithm was also very competitive in frequency-assignment problems [7, 24, 28]. Petford and Welsh experimented with their algorithm on a model of random 3-colorable graphs. They observed that there are some combinations of parameters of random graphs that are extremely hard for their algorithm, while their algorithm otherwise runs in linear time, on average. Having no theoretical explanation, Petford and Welsh write that the curious behavior "is not unlike the phenomenon of phase transition that occurs in the Ising model, Potts model and other models of statistical mechanics".

We have recently designed an algorithm for clustering that is motivated by this coloring algorithm. The results were very promising [14]. This motivated us to better understand the basic algorithm, which we did in an experimental study that is reported here. As high inherent parallelism is an interesting feature of the approach, we implemented a parallel version of the algorithm, keeping in mind its potential use in alternative models of computation. The algorithm was tested on a large quantity of data so as to have a better understanding of its performance. As the parallel execution is simulated on a sequential machine, the computation time is naturally measured in the number of parallel steps. In particular, we run and analyse the following experiments:

- In the experiments reported in [25, 26], the algorithms perform poorly in some instances. We perform several experiments and test the hypothesis that the performance of the algorithm depends only on the average degree for random graphs and for regular graphs.
- To test whether the performance of the algorithm depends on the average degree, we test the performance by varying the average degree of the graph and the temperature (parameter T) of the system. We test the performance on random graphs and on r-regular graphs (both are known to be colorable).
- We test whether the algorithm behaves in a similar way for higher-degree colorings ($k = 4, 5, \ldots, 10$) and confirm the existence of critical regions that are related to the phase transition phenomena.

The rest of the paper is organized as follows. In Section 2 we recall the k-coloring decision problem and the algorithms described in [25] and [26]. In Section 3 we present the new experiments and analyse the results. For our experiments we use two classes of randomly generated k-colorable graphs.

2. Graph coloring problem and the algorithm of Petford and Welsh

Graph coloring problem. The k-coloring decision problem (for $k \geq 3$) is a well-known NP-complete problem. It is formally stated as follows:

Input: graph G, integer k *Question: is there a proper k-coloring of G?*

A *coloring* is any mapping $c : V(G) \to \mathbb{N}$ and is a feasible solution of the k-coloring problem. A mapping $c : V(G) \to \mathbb{N}$ is a *proper coloring* of G if it assigns different colors to adjacent vertices. The cost function $E(c)$ is the number of *bad* edges. By definition, the bad edges for coloring c are edges with both ends colored by the same color in coloring c. Such vertices are called *bad*. Proper colorings are exactly the colorings with $E(c) = 0$ and finding a coloring c with $E(c) = 0$ is equivalent to answering the above decision problem. The coloring constructed is a *witness* c proving the correctness of the answer. A graph for which a proper coloring with k colors exists is said to be *k-colorable*.

The algorithm of Petford and Welsh. The basic algorithm [20] starts with an initial 3-coloring of the input graph that is chosen at random. Then an iterative process is started. During each iteration, a bad vertex is chosen at random. The chosen bad vertex is recolored randomly, according to some probability distribution. The color distribution favors colors that are less represented in the neighborhood of the chosen vertex, see the expression (1) below. The algorithm has a straightforward generalization to k-coloring (taking $k = 3$ gives the original algorithm) [27].

In pseudo code, the algorithm of Petford and Welsh is written as follows

Algorithm 1 Petford-Welsh algorithm

```
1: color vertices randomly with colors 1, 2, ..., k
2: while not stopping condition do
3:     select a bad vertex v (randomly)
4:     assign a new color c to v
5: end while
```

A bad vertex is selected uniformly at random among vertices that are endpoints of some bad (e.g., monochromatic) edge. A new color is assigned at random. The new color is taken from the set $\{1, 2, \ldots, k\}$. Sampling is conducted according to the probability distribution defined as follows:

The probability p_i of color i to be chosen as a new color of vertex v is proportional to

$$p_i \approx \exp(-S_i/T), \tag{1}$$

where S_i is the number of edges with one endpoint at v and with color i assigned to the other endpoint. Petford and Welsh used 4^{-S_i} which is equivalent to using $T \approx 0.72$ in (1). (Because $\exp(-x/T) = 4^{-x}$ implies $T \approx 0.72$.)

The stopping criterion includes two conditions. The algorithm stops either when the time limit (in terms of the number of calls to the function that computes a new color) is reached, or when a proper coloring is found. If a proper coloring is found, the answer to the decision problem is positive, and the proper coloring is reported as a witness. In the case when no proper coloring is found, the feasible solution with minimal cost $E(c)$ is reported as an approximate solution to the problem. The answer to the decision problem is in this case negative; however, it might not be correct. Note that there is no guarantee of the quality of the solution.

Connection to simulated annealing and the generalized Boltzmann machine. Here we briefly discuss the parameter T of the algorithm. Due to an analogy between the temperature of the simulated annealing algorithm and the temperature of the generalized Boltzmann machine neural network [22], parameter T can naturally be called the temperature. With the term *simulated annealing* (SA) we refer to the optimization heuristics as proposed, for example, in [16]. The *Generalized Botzmann machine* is a generalization of the Boltzmann machine, a neural network that is based on a stochastic spin-glass model and has been widely used in artificial intelligence [13]. The main difference between the two is that the generalized Boltzmann machine as defined in [22] uses multistate neurons, in contrast to the bipolar neurons of the usual Boltzmann machine.

These analogies are based on the following simple observation. Pick a vertex and denote the old color of the chosen vertex by j and the new color by i. The number of bad edges E' after the move is

$$E' = E - S_j + S_i \tag{2}$$

where E is the number of bad edges before the change. We define $\Delta E = E - E' = S_j - S_i$. During each step, j is fixed and hence S_j and E are fixed. Consequently, it is equivalent if we define the probability of choosing the new color i to be proportional to either $\exp(-S_i/T)$, $\exp(\Delta E/T)$ or $\exp(E'/T)$.

To see the relation to the Boltzmann machine, recall that the number of bad edges is a usual definition of the energy function, in both the simulated annealing and in the generalized Boltzmann machine with multistate neurons. This indicates that the algorithm of Petford and Welsh is closely related to the generalized Boltzmann machine operating at a constant temperature (for details, see [22] and the references therein). The major difference between the two is in the firing rule. While in the Boltzmann machine all the neurons are fired with equal probability, in the algorithm of Petford and Welsh, only the bad vertices are activated.

As already explained, the original algorithm of Petford and Welsh uses probabilities proportional to 4^{-S_i}, which corresponds to $T \approx 0.72$. Other choices of T are possible. On one hand, low values of T make the algorithm behave very much like an iterative improvement, and it will be quickly converging to a local minimum. On the other hand, a large T means a higher probability of accepting a move that increases the number of bad edges. Consequently, a very high T results in chaotic behavior that is similar to a pure random walk among the colorings, regardless of their energy.

Both Petford and Welsh and simulated annealing are local-search-type heuristics. Besides the different uses of T (fixed temperature versus cooling schedule), there is another slight difference. Namely, in the usual implementation of SA, a change that improves the cost is always accepted, while in the other case, the acceptance probability is used, which depends on both the difference in costs and the temperature. In the algorithm of Petford and Welsh, all the changes are made according to the probabilities using (1). The cost-improving changes thus might not be accepted, although this happens very rarely in the majority of cases.

Thus, our algorithm is similar to both the simulated annealing heuristics and to the sequential operation of the Boltzmann machine. Its acceptance probability is (for a given T) practically equivalent to the Boltzmann machine. As the Boltzmann machine is a highly parallel asynchronous device, a comparison with parallel implementations is even more interesting. Here, we recall the parallel version of the algorithm of Petford and Welsh that differs from the generalized Boltzmann machine only in the firing rules in both phases of its operation.

Parallel algorithm. In [25], a massively parallel version of Algorithm 1 was proposed. The naive algorithm (Algorithm 2) was later improved in [26] by a version that runs in two phases, thus avoiding the looping that can appear for some configurations within the instance. The improved algorithm (Algorithm 3) first aims to find a $2k$ coloring and does not recolor all the bad vertices simultaneously because each change is only done with some probability (i.e.,

0.6). In the second phase, the result of the first phase provides independent sets of vertices that can be recolored in parallel without any conflict. The resulting algorithm still shows the maximum speedup in comparison to the original version, e.g., the instances solved in linear time sequentially are expected to be solved in constant time in parallel [26]. The algorithm formally reads as Algorithm 3.

Note that in the first phase, the firing rule is nearly equivalent to that of the Boltzmann machine in which each neuron (vertex) wakes up at some random time and performs the recoloring. Note that there is no synchronization among the neurons. While synchronization is not of particular importance in the first phase of our algorithm, in the second phase, it is essential that synchronization based on the result of the first phase is used.

Algorithm 2 Massively parallel variant of the Petford–Welsh algorithm (naive version)

```
1: color vertices randomly with colors 1, 2, ..., k
2: while not stopping condition do
3:     bad_vertices ← {v | visbad}
4:     for all v ∈ bad_vertices do                    } in parallel
5:         assign a new color c to v                  }
6:     end for                                        }
7: end while
```

Algorithm 3 Massively parallel variant of the Petford–Welsh algorithm

```
1: procedure MPPW_PHASE1(G)
2:     color vertices randomly with colors 1, 2, ..., k, k + 1, ..., 2k
3:     while not stopping condition do
4:         bad_vertices ← {v | visbad}
5:         for all v ∈ bad_vertices do                                  } in parallel
6:             assign a new color c to v with a probability of 60%      }
7:         end for                                                      }
8:     end while
9: return coloring c
10: end procedure

11: procedure MPPW_PHASE2(G)
12:     bc ← MPPW_PHASE1(G)
13:     color vertices randomly with colors 1, 2, ..., k
14:     while not stopping condition do
15:         bad_vertices ← {v | visbad}
16:         for all v ∈ bad_vertices do                    } in parallel
17:             if step mod (2k) = bc(v) then              }
18:                 assign a new color c to v              }
19:             end if                                     }
20:         end for                                        }
21:     end while
22: return coloring c
23: end procedure
```

3. Experiments

In our experiments we use two classes of graphs. The first class is graphs of the form

$$G(n, k, p), \tag{3}$$

where n is the number of vertices, k is the number of partitions and p is the probability that two vertices from distinct partitions are adjacent (see [20, 25, 10]).

The second class is d-regular graphs of the form

$$R(n, k, d), \tag{4}$$

where n is the number of vertices, k is the number of partitions and d is the degree of vertices. The Python code for generating such graphs can be found in [10]. In short, the algorithm splits the vertices into k partitions and connects, in a random order, each vertex to d randomly chosen vertices in distinct partitions. If there are vertices left that are not of degree d and cannot be connected, we delete an edge and add two other edges. More precisely, if u and v are vertices with $\deg(u) < d$ and $\deg(v) < d$, then we find an edge $u'v'$, such that u and u' do not belong to the same partition and v and v' do not belong to the same partition. We delete the edge $u'v'$ and add edges uu' and vv' to the graph.

In both cases the partitions are of equal size if k divides n, otherwise their sizes differ by at most one vertex.

Preliminary experiment. With the code in [10] we reproduced the results from [25, 26], with a much larger sample size, $n = 10000$ (instead of $n = 100$). The results are presented in Figure 1, confirming that our implementation runs exactly the same algorithm as the original.

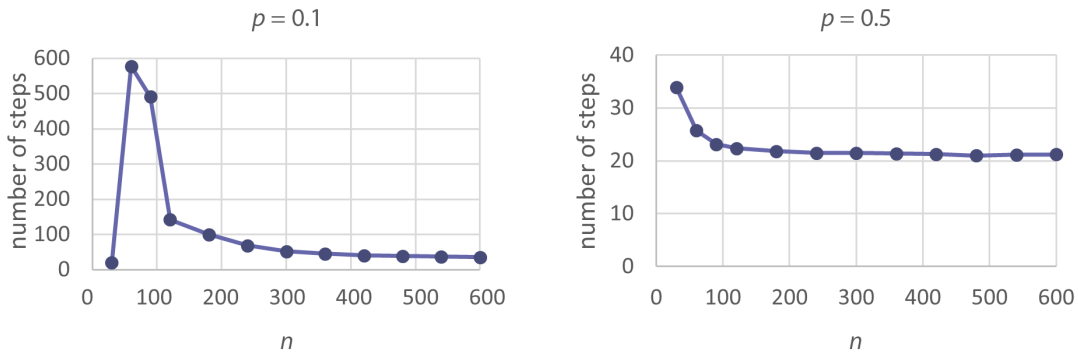


Figure 1: *Performance of IMPPW (parallel steps as a function of n), see Table 1 that resembles Table 1 in [25].*

$p = 0.1$		
n	$T_2(n)$	succ. runs
30	19.	100
60	261.	89
90	345.	93
120	142.	100
180	99.	100
240	71.	100
300	52.	100
360	45.	100
420	40.	100
480	39.	100
540	36.	100
600	35.	100
660	34.	100
720	34.	100
780	32.	100
840	33.	100
900	32.	100

$p = 0.5$		
n	$T_2(n)$	succ. runs
30	33.	100
60	25.	100
90	24.	100
120	23.	100
180	22.	100
240	22.	100
300	21.	100
360	21.	100
420	21.	100
480	21.	100
540	21.	100
600	22.	100
660	21.	100
720	22.	100
780	21.	100
840	21.	100
900	21.	100

Table 1: *Performance of IMPPW (number of parallel steps as a function of n). Number of successful runs (out of 100) is given in the third column.*

Observe that the algorithm does not perform well on a narrow interval only, which we call the *critical region*. Observing some experimental results, limited by the computing resources available at the time, the following conjecture was proposed [27].

Conjecture 1. *The critical regions are characterized by the equation*

$$\frac{2pn}{k} \approx \frac{16}{3}. \tag{5}$$

This conjecture generalizes the conjecture of Petford and Welsh, who observed that the equation $\frac{2pn}{3} \approx \frac{16}{3}$ is valid within the critical region [20]. More precisely, they observed that given p, the graphs $G(n, 3, p)$ with $n \approx \frac{8}{p}$ are likely hard instances for the algorithm.

After we confirm the basic observations in the main references, we continue with experimental results that can shed some more light on the behaviour of the algorithm and possibly on some more general phenomena. In particular, we wanted to understand better, if and how the hard instances can be related to the average degree of the graphs. In relation to this, we wish to check whether the conjecture above captures the main information that determines the critical regions. Furthermore, we are interested in the question "what is the effect of the temperature" (or, equivalently, the basis of the exponent expression (1)) on the performance of the algorithm? Below we provide the results of the experiments with some comments that answer some questions and, at the same time, highlight some new questions.

First experiment. In the first experiment we show that, with a fixed number of components, the critical region depends on the average degree $\overline{d}(G)$ of the graph, where

$$\overline{d}(G) = \frac{np(k-1)}{k}. \tag{6}$$

We choose four datasets: graphs of classes $G(90, 3, p)$, $G(120, 3, p)$, $G(300, 3, p)$, and $G(3000, 3, p)$. The sample size is 10000 for $n \in \{90, 120, 300\}$ and a bit smaller, 2000, for $n = 3000$. We choose the parameter p in such a way that the average degree of each class varies from 2 to 9. The results are presented in diagrams in Figure 2.

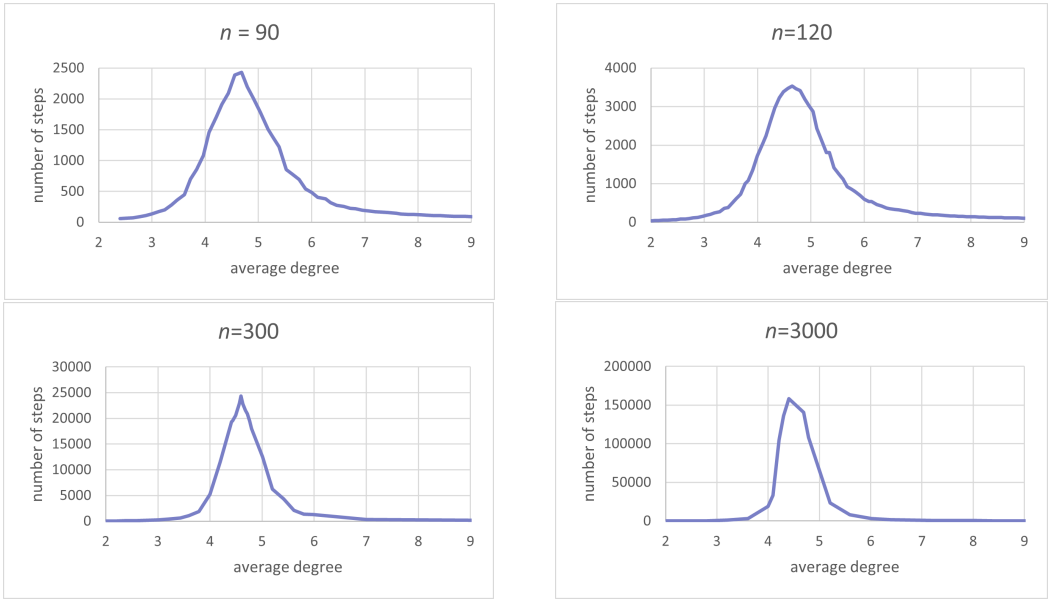


Figure 2: *Performance of IMPPW (steps as a function of average degree).*

Note that in all four experiments, the extreme value does not depend on the number of vertices. It is obvious that the hard instances are in the interval where the average degree is between 4 and 5; even more, the peaks are within the range 4.5–4.7 in all four diagrams.

The phenomena can be explained as follows. Observe that before the critical region ($\overline{d}(G) < 4.5$), the partitions (e.g., sets of vertices of the same color in a proper coloring) are loosely defined and there are multiple optimal solutions (see Supplementary Video 1). On the other hand, after the critical region ($\overline{d}(G) > 4.7$), the algorithm converges fast, since the partitions are densely connected and thus very well defined (see Supplementary Video 3). For a graph in the critical region see Supplementary Video 2. All three videos can also be accessed at [10].

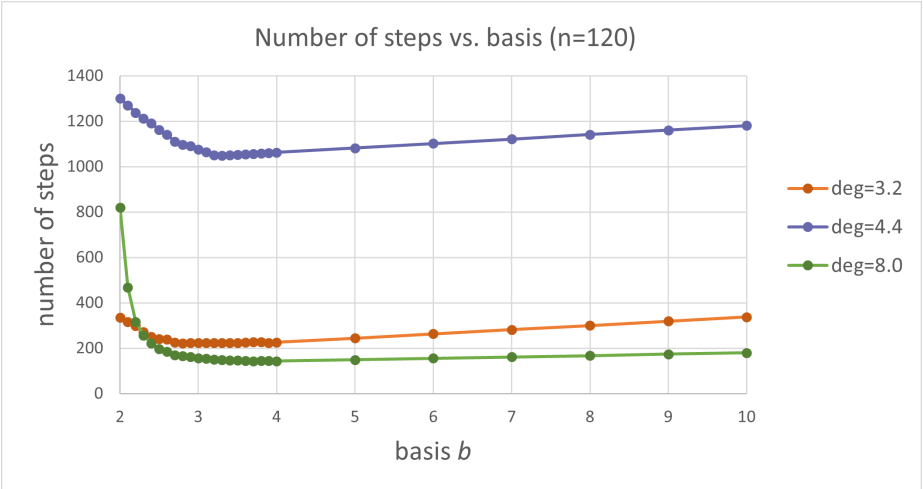


Figure 3: *Performance of IMPPW (basis vs. number of steps).*

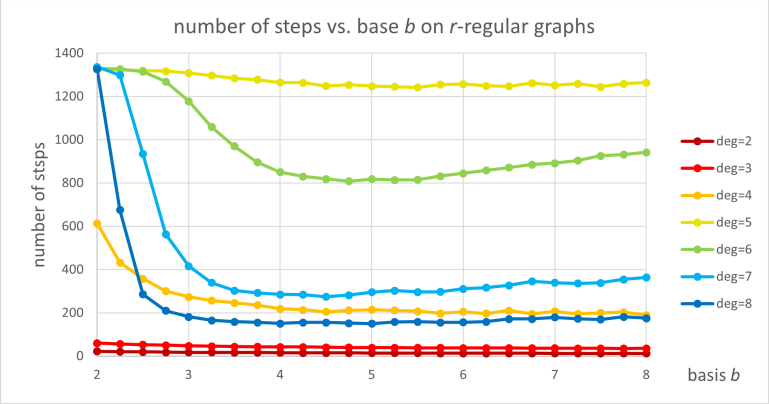


Figure 4: *Performance of IMPPW (basis vs. number of steps for d-regular graphs).*

Second experiment. In this experiment, we vary the parameter b over the values from 2 to 10 and measure the performance of the algorithm in the classes $G(120, 3, p)$, where p is chosen in such a way that the average degrees are 3.2, 4.4, and 8.0. The results are presented in the diagram in Figure 3.

We conclude that the basis b does not dramatically influence the performance. Seemingly, the values between 3.0 and 4.0 are well behaved and indeed, taking any value $b \in [3, 4]$, perform similarly. We repeat the experiment on the graphs $R(120, 3, d)$, where $d = 2, 3, \ldots, 8$. The results are presented in the diagram in Figure 4 and clearly confirm the earlier observations.

The question "what is the optimal value b?" should have an answer between 3 and 4. Note that this is a question equivalent to the question as to which is the optimal temperature of the simulated annealing (i.e. "annealing" with constant temperature)? However, this seems to be a rather complex problem [4, 29]. (Recall that $\exp(-x/T) = b^{-x}$ so $b = \exp(1/T)$.) As our insight is limited by the special class of instances used, we do not wish to dig deeper into the question of optimal b. On the other hand, we can confirm that, probably, the algorithm is robust regarding the choice of b and, equivalently, to the choice of parameter T).

Third experiment. In this experiment we compute the run times for the graphs in $G(n, k, p)$ where we fix $k = 3, 4, \ldots, 8$ partitions and $n = 60, 120, 240$ vertices, and in each case observe how the number of steps needed depends on the probability p. In all the experiments the sample size is 5000. The results are presented in Figure 5.

According to Conjecture 1, the hard instances are characterized by some constant value of $\frac{2pn}{k}$. However, in Figure 6 we observe how the performance depends on $\frac{2pn}{k}$. We conclude that the critical region is not characterized exactly by $\frac{2pn}{k}$, thus the Conjecture 1 should be replaced by a better one.

In Figure 7 we plot how the number of steps depends on the expression $\frac{p(k-1)}{k}$, which is proportional, if n is fixed, to the average degree. We observe from the figure that the critical region cannot be explained only by the average degree.

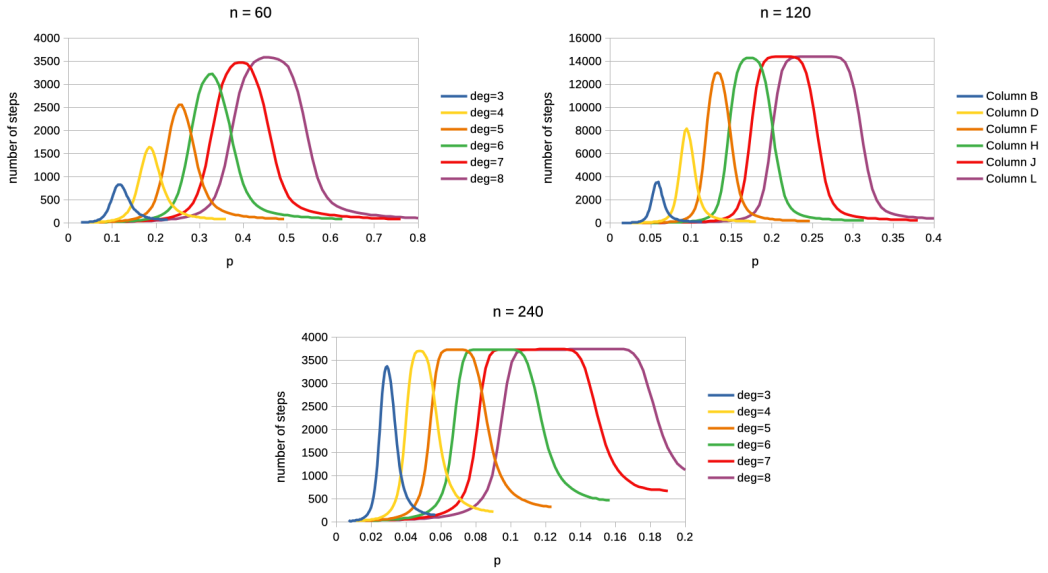


Figure 5: *Performance of IMPPW for k-colorings. The graphs show how the number of steps depends on the parameter p.*

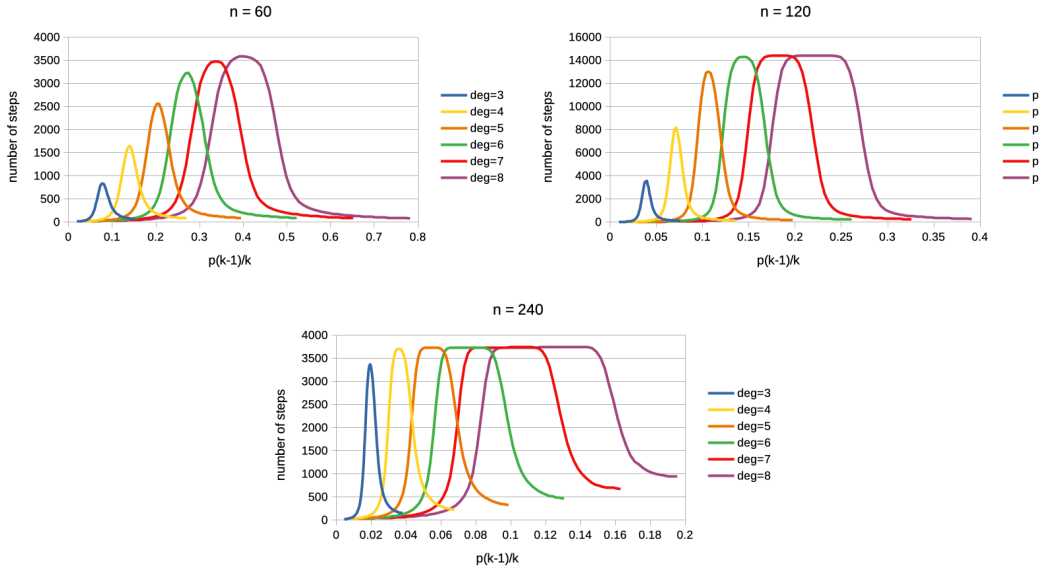


Figure 7: *Performance of IMPPW for k-colorings. The graphs show how the number of steps depends on the average degree, which is proportional to* $\frac{p(k-1)}{k}$.

At present we do not have a good idea of how to improve the conjecture to better characterize the critical region with an expression that would have some natural meaning. We have tested some slight modifications and found that the expression $\frac{p}{k-1.5}$ remains fairly constant for varied k. See Figure 8, where we plot how the number of steps depends on the expression $\frac{p}{k-1.5}$.

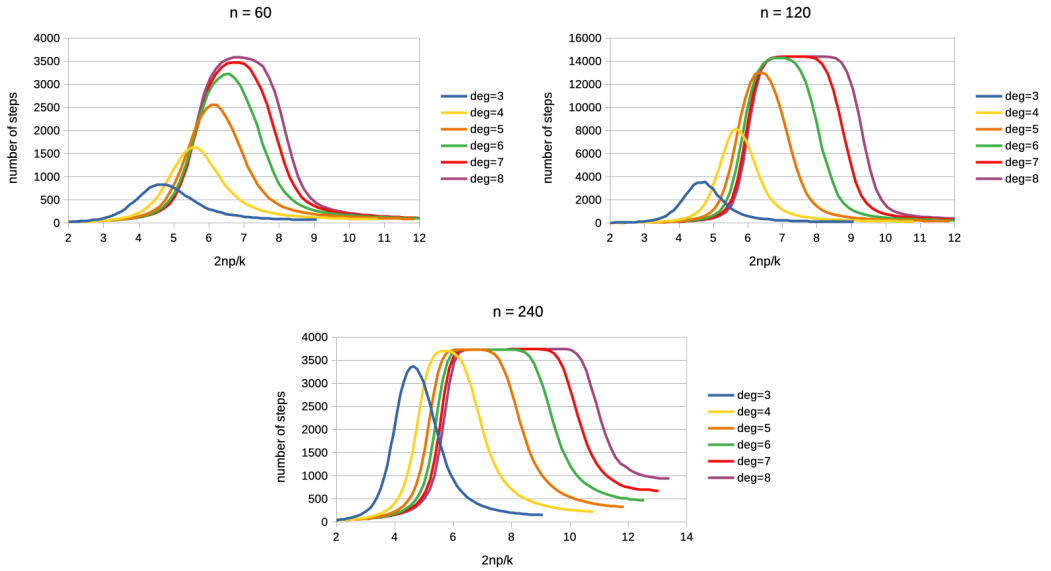


Figure 6: Performance of IMPPW for k-colorings. The graphs show how the number of steps depends on the parameter p.

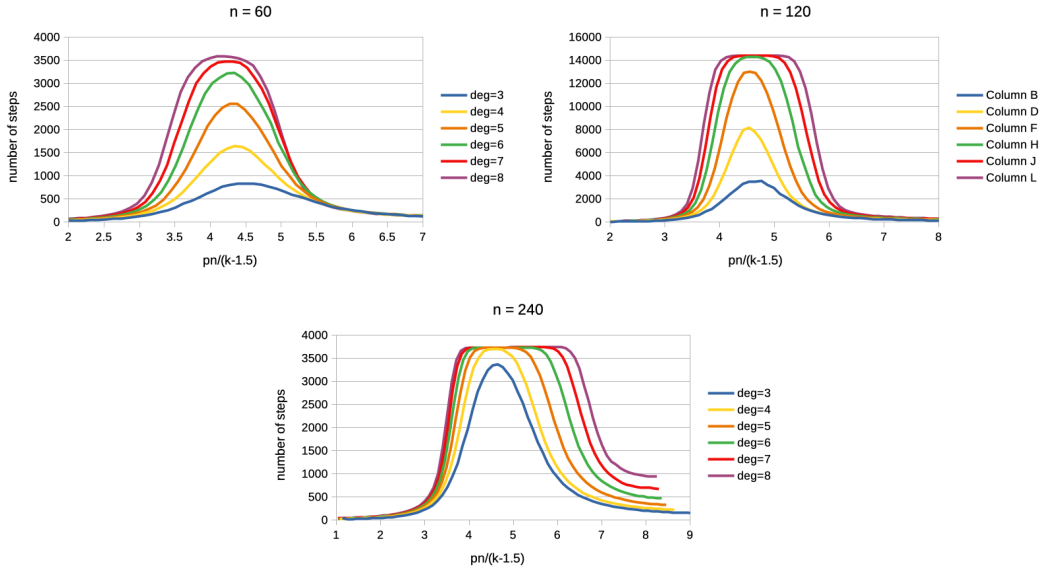


Figure 8: *Performance of IMPPW for k-colorings. The graphs show how the number of steps depends on the expression* $\frac{pn}{k-1.5}$. *The cut-off is due to the time constraint of the algorithm.*

Therefore, it seems that the critical region is approximately characterized by the equation

$$\frac{np}{k-1.5} \approx 4.3. \tag{7}$$

4. Conclusions

A parallel version of Petford and Welsh's k-coloring algorithm was extensively tested on two classes of random graphs. The conjecture about the existence of a critical region where the algorithm has nearly prohibitively long run times was confirmed for the case $k = 3$, while a generalized conjecture [27] was shown to need an adjustment. More precisely, we have observed that the critical region appears where $\frac{p}{k-1.5}$ holds. This result opens at least two interesting questions.

- can the property $\frac{np}{k-1.5} \approx 4.3$ be naturally explained as some feature of the instances?
- is there another expression that fits the data, and has some meaning that can explain the structure of hard instances?

In contrast to the graph classes of k colorable graphs used in this paper, the usual random graph model considered are graphs $G(n, p)$ where each of the possible $\frac{n(n-1)}{2}$ edges appears independently with a probability p. Not surprisingly, for 3-coloring, it is found that the critical mean degree where the phase transition occurs is around $\alpha_{crit} \approx 4.7$, for example, the estimate 4.703 was put forward in [2]. In the same paper, the analysis implies that the hardest instances are among the graphs with an average degree between 4.42 and α_{crit}. It seems that the k-coloring was not considered, hence we cannot compare our findings about the hardest instances with previous work. We conclude that further study of the critical regions is a promising avenue of research that might have some implications that go beyond understanding of the behavior of the algorithm of Petford and Welsh.

Acknowledgements

The authors were supported in part by the Slovenian Research Agency, grants J2-2512, J1-4031, N1-0278 (B. Gabrovšek), and J2-2512 and P2-0248 (J. Žerovnik).

References

[1] Babaei, H., Karimpour, J. and Hadidi, A. (2015). A survey of approaches for university course timetabling problem. Computers & Industrial Engineering, 86, 43-59. doi: 10.1016/j.cie.2014.11.010

[2] Boettcher, S. and Percus, A. G. (2004) . Extremal optimization at the phase transition of the three-coloring problem. Phys. Rev. E, 69, 066703. doi: 10.1103/PhysRevE.69.066703

[3] Cipra, B. A. (2000). The Ising model is NP-complete. SIAM News, 33(6), 1-3. Retrieved from: semanticscholar.org

[4] Cohn, H. and Fielding, M. (1999) . Simulated annealing: searching for an optimal temperature schedule. SIAM Journal on Optimization, 9, 779-802. doi: 10.1137/S1052623497329683

[5] Culberson, J. and Gent, I. (2001). Frozen development in graph coloring. Theoretical Computer Science, 265, 227-264. doi: 10.1016/S0304-3975(01)00164-5

[6] Donnelly, P. and Welsh, D. (1984). The antivoter problem: random 2-colourings of graphs. In Graph Theory and Combinatorics (Cambridge, 1983), (pp. 133-144), Academic Press, London.

[7] Chamaret, B., Ubeda, S. and Žerovnik, J. (1996). A Randomized Algorithm for Graph Colouring Applied to Channel Allocation in Mobile Telephone Networks. Proceedings of the 6th International Conference on Operational Research KOI'96. 25-30, Croatian Operational Research Society, Zagreb.

[8] Dubois, O., Monasson, R., Selman, B. and Zecchina, R. (2001). Editorial. Theoretical Computer Science, 265(1–2), 1. doi: 10.1016/S0304-3975(01)00133-5

[9] Fortnow, L. (2009). The status of the P versus NP problem. Commun. ACM 52, 9 (September 2009), 78–86. doi: 10.1145/1562164.1562186

[10] Gabrovšek, B. (2023). Massively parallel algorithm for graph coloring based on the Petford-Welsh algorithm, source code, Retrieved from: github.com/bgabrovsek/petford-welsh-coloring

[11] Galinier, P. and Hertz, A. (2006). A survey of local search methods for graph coloring. Computers Operations Research, 33(9), 2547–2562. doi: 10.1016/J.COR.2005.07.028
[12] Garey, M. R. and Johnson, D. S. (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman.
[13] Hinton, G. (2011). Boltzmann Machines. In Sammut, C. and Webb, G.I. (Eds.) Encyclopedia of Machine Learning. Springer, Boston, MA. doi: 10.1007/978-0-387-30164-8$_8$3
[14] Ikica, B., Gabrovšek, B., Povh, J. and Žerovnik, J. (2022). Clustering as a dual problem to colouring. Computational and Applied Mathematics, 41, 147. doi: 10.1007/s40314-022-01835-0
[15] Karp, R. (1972), Reducibility among combinatorial problems. In R. Miller and J. Thatcher (Eds.) Complexity of Computer Computations. Plenum Press, 85-103.
[16] Kirckpatrick, S., Gellat, C. D. and Vecchi, M. P. (1983). Optimization by simulated annealing. Science, 220(4598), 671-680. doi: 10.1126/science.220.4598.671
[17] Lewis, R. M. R. (2021). Guide to Graph Colouring, Texts in Computer Science (second edition), Spirnger Nature Switzerland. doi: 10.1007/978-3-030-81054-2
[18] Martín, H. J. A. (2013). Solving Hard Computational Problems Efficiently: Asymptotic Parametric Complexity 3-Coloring Algorithm. PLoS ONE, 8(1), e53437. doi: 10.1371/journal.pone.0053437
[19] Mézard, M. (2022). Spin glasses and optimization in complex systems. Europhysics News, 53(1), 15-17. doi: 10.1051/epn/2022105
[20] Petford, A. and Welsh, D. (1989). A Randomised 3-coloring Algorithm. Discrete Mathematics, 74, 253-261. doi: 10.1016/0012-365X(89)90214-8
[21] Philathong, H., Akshay V., Samburskaya, K. and Biamonte, J. (2021). Computational phase transitions: benchmarking Ising machines and quantum optimisers. Journal of Physics: Complexity, 2(1), 011002. doi: 10.1088/2632-072X/abdadc
[22] Shawe-Taylor, J., and Žerovnik, J. (1992). Boltzmann Machines with Finite Alphabet. In Proceedings International Conference on Artificial Neural Networks, ICANN'92. Elsevier Science. 391-394 . Retrieved from: eprints.soton.ac.uk
[23] Shawe-Taylor, J., and Žerovnik, J. (1995). Analysis of the Mean Field Annealing Algorithm for Graph Colouring. Journal of Artificial Neural Networks, 2, 329-340.
[24] Ubeda, S. and Žerovnik, J. (1997). A randomized algorithm for a channel assignment problem. Speedup, 11, 14-19.
[25] Žerovnik, J. (1990). A parallel variant of a heuristical algorithm for graph coloring. Parallel Computing, 13, 95-100.
[26] Žerovnik, J. and Kaufman, M. (1992). A parallel variant of a heuristical algorithm for graph coloring - corrigendum. Parallel Computing, 18, 897-900.
[27] Žerovnik, J. (1994). A Randomized Algorithm for k-colorability. Discrete Mathematics, 131, 379-393. doi: 10.1016/0012-365X(94)90402-2
[28] Žerovnik, J. (1998) . On the convergence of a randomized algorithm frequency assignment problem. Central European Journal for Operations Research and Economics, 6, 135-151.
[29] Žerovnik, J. (2000). On temperature schedules for generalized Boltzmann machine. Neural Network World, 3, 495-503.

On some quantitative indicators of a few features of electoral systems

Tomislav Marošević[1,*] **and Josip Miletić** [2]

[1] *School of Applied Mathematics and Informatics, Josip Juraj Strossmayer University of Osijek, Trg Ljudevita Gaja 6, HR-31000 Osijek, Croatia*
E-mail: ⟨tmarosev@mathos.hr⟩
[2] *Faculty of Electrical Engineering, Computer Science and Information Technology, Josip Juraj Strossmayer University of Osijek, Kneza Trpimira 2 B, HR-31000 Osijek, Croatia*
E-mail: ⟨josip.miletic@ferit.hr⟩

Abstract. Electoral systems can be analyzed by means of their numerous properties. Some quantitative indicators of a few technical features of electoral systems are considered. With respect to the effective number of parties in a electoral system, one can observe some known indicators (e.g., fractionalization of vote shares, the Laakso-Taagepera index, the Wildgen index, the Molinar index). With regard to the government stability, one can look at the indicator which is called the expectation of government stability. These indicators are examined from the empirical point of view, i.e., in relation of elections in different countries.

Keywords: electoral systems, government stability indicator, Laakso-Taagepera index, vote fractionalization, Wildgen index

Received: February 18, 2024; accepted: August 12, 2024; available online: October 7, 2024

DOI: 10.17535/crorr.2024.0010

1. Introduction

Electoral systems can be analyzed by means of their numerous properties. For instance, electoral participation, number of parties [12] proportionality [18], party power and coalitions [20], government stability [9] can be observed. These features of electoral systems have certain influence on the electoral outcome and on the political structure of the country.
The number of parties in an electoral system is important, because it describes a basic variable of the electoral system. Several quantitative indicators of the effective number of parties that refer to the phase of electoral process before the transformation of votes into seats can be considered [4]. In this paper some indicators in relation to the empirical cases of elections in different countries are considered. Some indicators of effective number of parties are sensitive to the formal count n of parties [7]. The contribution of this paper to the reduction of this sensitivity is through modification of these indicators which is suggested by elimination of tiny parties. These modified indicators are calculated in several cases of elections in various countries and compared with corresponding indicators (the idea of not taking into account tiny parties has been used with Rae disproportionality indicator [19]). Some recent research about the effective number of parties can be seen in [8, 13, 22].

The government formation appears in electoral phase which is after the transformation of votes into seats. There are various approaches to this complex problem [9, 11, 21]. Numerous factors can influence the government stability (e.g., image of competence, government provided staff, media exposure, see [17, 10], and some recent research can be seen in [1, 3, 5]. With

*Corresponding author.

regard to the government stability, a special indicator which is called the expectation of government stability can be observed [4]. In the paper, this indicator is viewed from the empirical standpoint by a few examples of elections in several countries. The formula of the indicator of the government stability expectation uses the so-called minimal winning coalitions with distance threshold [4]. However, in certain practical situation the minimal winning coalitions with distance threshold can be formed with a very small possibility. This paper contributes by the modification of this indicator, by using the so-called politically feasible winning coalitions [15] that can better correspond real situation of government formation. In several empirical cases of various parliaments, the modified expectation of government stability is calculated and compared with the indicator of government stability expectation.

In Subsection 1.1, some well-known indicators of the effective number of parties are described.

1.1. The number of political parties

The formal number n of parties that go to elections can be just counted, but it does not seem to be the best way to describe complex relationships between the parties and their relative strength. The effective number of parties is used as an important parameter to illustrate competitive relation between the various actors of the political system. For example, if the total number of parties that participate in the election is $n \geq 3$ where two parties have approximately equal number of votes and almost all the votes together, then that system can be considered as a two-party system.

Therefore, with respect to the effective number of parties in the system, the votes each party obtains are taken into account by many proposed indicators. They refer to the pretransformation phase of the electoral process, i.e., before the transformation of votes into seats. The following notation is used:

- $n \geq 2$ - the total number of parties in the elections
- $v_i > 0$ - the number of votes that party i has got in the election ($i = 1, \ldots, n$), and $P = \sum_{i=1}^{n} v_i$
- $\omega_i = \frac{v_i}{P}$ - vote share of the party i ($i = 1, \ldots, n$), where $\sum_{i=1}^{n} \omega_i = 1$.

General indicator N_α for the effective number of parties in the electoral system, where vote shares ω_i is considered, is the following:

$$N_\alpha = (\sum_{i=1}^{n} \omega_i^\alpha)^{\frac{1}{1-\alpha}}, \tag{1}$$

where α is parameter.

Notice that for $\alpha = 2$, from (1) one gets the Laakso-Taagepera index [12]

$$N_2 = \frac{1}{\sum_{i=1}^{n} \omega_i^2}. \tag{2}$$

Notice that the vote share of the party i, $\omega_i = \frac{v_i}{P}$, can be viewed as the probability that one randomly selected voter votes for the party i. Thus, quantity in the denominator of (2), $\sum_{i=1}^{n} \omega_i^2$,

represents the probability that two randomly selected voters vote for the same party.

Reciprocally, the probability that two randomly selected voters do not vote for the same party is the following:

$$1 - \sum_{i=1}^{n} \omega_i^2 = F_2, \tag{3}$$

where F_2 is called the vote fractionalization [18]. It holds that $F_2 \in [0, 1]$.

From (2) and (3) it follows that

$$F_2 = 1 - \frac{1}{N_2}\,. \tag{4}$$

When parameter α tends to 1, then from (1) one obtains the so-called Wildgen index N_1 [24]:

$$N_1 = \lim_{\alpha \to 1} N_\alpha = e^{-\sum_{i=1}^{n} \omega_i \ln \omega_i}\,. \tag{5}$$

Notice that when parameter α tends to 0, then the general indicator N_α by (1) tends to n.

It can be seen that Laakso-Taagepera indicator N_2 is too sensitive to the share of votes of the largest party. The Wildgen indicator N_1 is more sensitive than the others to the total number of parties, even if they are tiny. In order to improve these imperfections, the Molinar indicator (denoted by NP) was introduced in the following form [16]:

$$NP = 1 + \frac{\sum_{i=1}^{n} \omega_i^2 - \omega_{[1]}^2}{(\sum_{i=1}^{n} \omega_i^2)^2} = 1 + N_2 - N_2^2 \cdot \omega_{[1]}^2, \tag{6}$$

where $\omega_{[1]}$ is the largest vote share. The Molinar indicator uses Laakso-Taagepera indicator N_2 and separates the largest party.

The following properties hold: $NP \leq N_2 \leq N_1$. Further, when $\omega_i = \dfrac{1}{n}$, $\forall i = 1, \ldots, n$, then $NP = N_2 = N_1 = n$.

In Section 2, some well-known indicators of the effective number of parties are examined by the cases of elections for the European Parliament in the EU member states, and modification of these indicators is suggested. In Section 3 the indicator of the expectation of government stability is considered in several empirical cases of parliaments after elections in certain countries in the EU and its modification is suggested. Finally, in Section 4 a few concluding remarks are given with respect to the suggested modified indicators which provide certain improvements compared to the original indicators.

2. Example for the EU states. Modified indicators of the number of parties.

Example 1. *In this example indicators N_2, N_1 and NP have been calculated in the cases of elections for the European Parliament in 2004, 2009, 2014, 2019 and 2024 in every member state of the European Union (EU). All member states of the EU use proportional electoral system in these elections [6].*

Corresponding values of these indicators per EU member state are given in Table 1. (Since 2020, the United Kingdom is not member of the EU.)

St.	F_2	N_2	N_1	NP	n
Ge	*0,791226*	*4,79*	*7,17*	*2,73*	*24*
	0,824429	*5,70*	*8,28*	*3,65*	*32*
	0,808839	*5,23*	*7,66*	*3,76*	*25*
	0,85836	*7,06*	*10,11*	*5,52*	*41*
	0,871574	*7,79*	*11,06*	*5,38*	*35*
Fr	*0,845679*	*6,48*	*8,12*	*3,97*	*11*
	0,857488	*7,02*	*9,82*	*5,34*	*34*
	0,848311	*6,59*	*8,17*	*4,90*	*12*
	0,846433	*6,51*	*8,38*	*4,22*	*12*
	0,837767	*6,16*	*8,94*	*3,40*	*37*
It	*0,833528*	*6,01*	*9,59*	*3,52*	*25*
	0,782984	*4,61*	*6,52*	*2,97*	*16*
	0,752989	*4,05*	*5,51*	*2,36*	*12*
	0,787954	*4,72*	*6,37*	*3,11*	*18*
	0,823983	*5,68*	*7,40*	*4,01*	*15*
(U)	*0,833457*	*6,00*	*8,61*	*4,58*	*32*
	0,847336	*6,55*	*9,47*	*4,34*	*31*
	0,80597	*5,15*	*7,46*	*4,28*	*30*
	0,824739	*5,71*	*8,13*	*3,74*	*23*
Sp	*0,631421*	*2,71*	*3,56*	*2,30*	*31*
	0,659211	*2,93*	*4,18*	*2,37*	*35*
	0,844335	*6,42*	*9,77*	*4,48*	*39*
	0,81255	*5,33*	*7,45*	*3,2*	*32*
	0,770479	*4,35*	*6,56*	*3,10*	*34*
Pl	*0,866993*	*7,52*	*9,58*	*5,24*	*21*
	0,70595	*3,40*	*4,61*	*2,12*	*12*
	0,773091	*4,41*	*5,90*	*3,40*	*12*
	0,638856	*2,77*	*3,48*	*2,19*	*9*
	0,708468	*3,43*	*4,18*	*2,81*	*11*
Ro	*0,829278*	*5,86*	*7,976*	*4,01*	*14*
	0,77561	*4,46*	*5,31*	*3,54*	*8*
	0,800973	*5,02*	*7,36*	*2,46*	*16*
	0,813011	*5,35*	*6,84*	*4,26*	*14*
	0,720669	*3,58*	*5,67*	*1,56*	*13*
Ne	*0,844502*	*6,43*	*8,14*	*4,96*	*15*
	0,871897	*7,81*	*8,93*	*6,36*	*17*
	0,888711	*8,99*	*10,16*	*8,05*	*19*
	0,887935	*8,92*	*10,54*	*7,04*	*16*
	0,88511	*8,70*	*10,94*	*6,33*	*15*
Be	*0,881528*	*8,44*	*9,90*	*7,28*	*22*
	0,906863	*10,74*	*12,64*	*9,34*	*30*
	0,896797	*9,69*	*11,19*	*7,97*	*15*
	0,907875	*10,85*	*11,90*	*9,49*	*18*
	0,903734	*10,39*	*11,67*	*9,03*	*18*
Gr	*0,686474*	*3,19*	*4,49*	*2,31*	*21*
	0,745303	*3,93*	*5,88*	*2,86*	*27*
	0,851304	*6,73*	*11,04*	*4,53*	*40*
	0,818273	*5,52*	*10,22*	*3,18*	*40*
	0,857895	*7,04*	*10,71*	*4,04*	*31*
Cz	*0,831157*	*5,92*	*8,52*	*3,76*	*31*
	0,819901	*5,55*	*8,88*	*3,50*	*33*
	0,892622	*9,31*	*12,28*	*8,06*	*38*
	0,879694	*8,31*	*11,31*	*6,21*	*39*
	0,846182	*6,50*	*9,14*	*4,61*	*30*
Sw	*0,848514*	*6,60*	*7,70*	*4,97*	*15*
	0,85808	*7,05*	*8,62*	*5,09*	*15*
	0,864653	*7,39*	*8,71*	*5,19*	*14*
	0,854204	*6,86*	*8,02*	*5,26*	*22*
	0,848483	*6,60*	*7,81*	*4,93*	*20*

St.	F_2	N_2	N_1	NP	n
Pr	*0,653222*	*2,88*	*3,76*	*2,10*	*13*
	0,769197	*4,33*	*5,53*	*3,17*	*13*
	0,76603	*4,27*	*5,89*	*3,16*	*16*
	0,790327	*4,77*	*6,98*	*2,84*	*17*
	0,767989	*4,31*	*5,87*	*3,32*	*17*
Hu	*0,647964*	*2,86*	*3,64*	*2,03*	*8*
	0,626287	*2,68*	*3,72*	*1,40*	*8*
	0,684297	*3,17*	*4,40*	*1,51*	*8*
	0,677473	*3,10*	*4,62*	*1,44*	*9*
	0,698806	*3,32*	*4,58*	*2,12*	*11*
Au	*0,741729*	*3,87*	*4,41*	*3,21*	*6*
	0,794289	*4,86*	*5,53*	*3,74*	*8*
	0,801197	*5,03*	*5,81*	*4,49*	*9*
	0,766852	*4,29*	*4,87*	*3,09*	*7*
	0,797473	*4,94*	*5,53*	*4,37*	*7*
Bu					
	0,853409	*6,82*	*8,28*	*5,05*	*14*
	0,8225	*5,63*	*7,84*	*3,70*	*22*
	0,815498	*5,42*	*7,78*	*3,75*	*23*
	0,862094	*7,25*	*10,12*	*5,14*	*31*
De	*0,816956*	*5,46*	*8,14*	*3,28*	*9*
	0,840611	*6,27*	*6,94*	*5,46*	*9*
	0,83283	*5,98*	*6,77*	*4,45*	*8*
	0,84957	*6,65*	*7,86*	*5,21*	*10*
	0,887333	*8,88*	*9,8*	*7,48*	*11*
Fi	*0,819971*	*5,55*	*7,74*	*4,82*	*14*
	0,845158	*6,65*	*7,55*	*5,21*	*14*
	0,853799	*6,84*	*7,88*	*5,45*	*15*
	0,861045	*7,20*	*8,36*	*5,96*	*18*
	0,848415	*6,60*	*7,69*	*4,92*	*14*
Sk	*0,86318*	*7,31*	*10,77*	*6,75*	*17*
	0,829075	*5,85*	*8,30*	*3,34*	*16*
	0,891787	*9,24*	*13,69*	*5,28*	*29*
	0,890687	*9,15*	*12,52*	*6,76*	*31*
	0,828227	*5,82*	*8,04*	*4,20*	*23*
Ir	*0,786302*	*4,68*	*5,96*	*3,77*	*7*
	0,807732	*5,19*	*6,06*	*3,90*	*8*
	0,817256	*5,47*	*6,47*	*4,98*	*11*
	0,826356	*5,76*	*7,01*	*3,86*	*13*
	0,869713	*7,68*	*9,98*	*6,13*	*16*
Cr					
	0,779832	*4,54*	*8,25*	*3,31*	*28*
	0,72322	*3,61*	*5,47*	*2,37*	*25*
	0,88493	*8,69*	*13,10*	*5,79*	*33*
	0,791779	*4,80*	*7,54*	*2,96*	*25*
Li	*0,842445*	*6,35*	*8,14*	*3,68*	*12*
	0,853568	*6,83*	*9,09*	*4,46*	*15*
	0,862893	*7,29*	*8,07*	*6,66*	*10*
	0,892238	*9,28*	*11,65*	*6,91*	*16*
	0,88685	*8,84*	*11,21*	*6,29*	*15*
La	*0,840644*	*6,28*	*8,97*	*3,71*	*16*
	0,864835	*7,40*	*9,83*	*5,04*	*17*
	0,731497	*3,72*	*5,59*	*1,72*	*14*
	0,844416	*6,43*	*8,15*	*4,55*	*16*
	0,852599	*6,78*	*9,16*	*4,82*	*16*
Sn	*0,83317*	*5,99*	*7,21*	*5,0*	*13*
	0,837013	*6,14*	*7,42*	*4,46*	*12*
	0,87335	*7,90*	*10,36*	*5,07*	*16*
	0,848132	*6,58*	*8,54*	*4,60*	*14*
	0,82428	*5,69*	*7,43*	*3,66*	*11*

Table 1a: *Indicators per the EU state in 5 electoral years.*

State	F_2	N_2	N_1	NP	n
Estonia	*0,793738*	*4,85*	*7,64*	*2,67*	*11*
	0,792391	*4,82*	*5,90*	*4,24*	*12*
	0,81064	*5,28*	*5,78*	*4,63*	*9*
	0,822843	*5,65*	*6,57*	*4,45*	*10*
	0,840226	*6,26*	*7,05*	*5,45*	*10*
Cyprus	*0,789211*	*4,74*	*5,59*	*3,95*	*10*
	0,724208	*3,62*	*4,48*	*2,95*	*9*
	0,75992	*4,17*	*5,59*	*2,69*	*11*
	0,800278	*5,01*	*6,35*	*3,90*	*14*
	0,828763	*5,84*	*7,11*	*4,75*	*14*
Luxembourg	*0,761747*	*4,20*	*4,92*	*2,77*	*7*
	0,79358	*4,84*	*5,62*	*3,54*	*8*
	0,81064	*4,73*	*6,14*	*2,56*	*9*
	0,822843	*6,25*	*7,13*	*5,45*	*10*
	0,835228	*6,07*	*7,34*	*5,14*	*13*
Malta	*0,598492*	*2,49*	*2,88*	*2,04*	*8*
	0,535367	*2,15*	*2,46*	*1,76*	*10*
	0,553163	*2,24*	*2,60*	*1,81*	*7*
	0,560188	*2,27*	*2,78*	*1,75*	*9*
	0,614569	*2,59*	*3,27*	*2,22*	*9*

Table 1b: *Indicators per member states of the EU in 5 electoral years.*

In Table 1a, 1b extreme values of corresponding indicators can be noticed and they are given in Table 2.

$\min N_2 = 2,15$	*Malta,*	*2009*	$\max N_2 = 10,85$	*Belgium,*	*2019*
$\min F_2 = 0,535367$			$\max F_2 = 0,907875$		
$\min N_1 = 2,46$	*Malta,*	*2009*	$\max N_1 = 13,69$	*Slovakia,*	*2014*
$\min NP = 1,40$	*Hungary,*	*2009*	$\max NP = 9,49$	*Belgium,*	*2019*
($\min n = 6$	*Austria,*	*2004)*	*(*$\max n = 41$	*Germany,*	*2019)*

Table 2: *Extreme values of indicators from Table 1.*

In order to summarize multitude data in Table 1, let us observe the average values of indicators per states in EU that are given in Table 3. Average values of the indicator N_2 of each EU member state are illustrated in Figure 1.

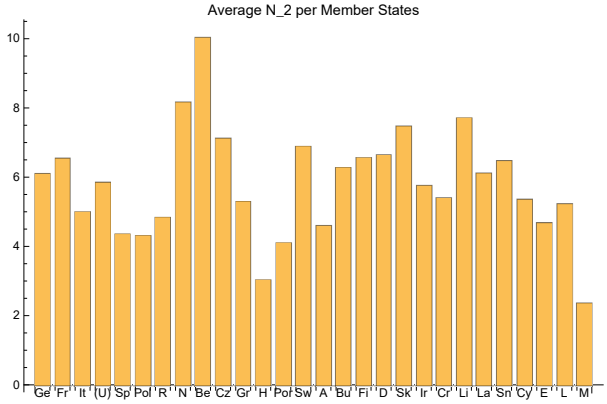


Figure 1: *Average values of the indicator N_2 of each EU member state.*

In order to summarize a lot of data in Figure 1, the average values of indicator N_2 per states in EU are rounded and put together by integer values in the groups: 2, 3, ...,8, ≥ 9. *These*

Indicator	$\bar{F}_2$	$\bar{N}_2$	$\bar{N}_1$	$\bar{NP}$	$\bar{n}$
Germany	0, 830886	6, 11	8, 86	4, 21	31, 4
France	0, 847136	6, 55	8, 69	4, 37	21, 2
Italy	0, 796288	5, 01	7, 08	3, 19	17, 2
(UK)	0, 827876	5, 85	8, 42	4, 23	29
Spain	0, 743599	4, 35	6, 30	3, 09	34, 2
Poland	0, 738672	4, 31	5, 55	3, 15	13
Romania	0, 787908	4, 85	6, 63	3, 17	13
Netherlands	0, 875631	8, 17	9, 74	6, 55	16, 4
Belgium	0, 893723	10, 02	11, 46	8, 62	20, 6
Czechia	0, 853911	7, 12	10, 03	5, 23	34, 2
Greece	0, 79185	5, 28	8, 47	3, 38	31, 8
Hungary	0, 666965	3, 03	4, 19	1.7	8, 8
Portugal	0, 749353	4, 11	5, 61	2, 92	15, 2
Sweden	0, 854787	6, 9	8, 17	5, 09	17, 2
Austria	0, 780308	4, 60	5, 23	3, 78	7, 4
Bulgaria	0, 838375	6, 28	8, 51	4, 40	22, 5
Finland	0, 845678	6, 57	7, 84	5, 27	15
Denmark	0, 84546	6, 65	7, 90	5, 18	9, 4
Slovakia	0, 860591	7, 47	10, 66	5, 27	23, 2
Ireland	0, 821472	5, 76	7, 10	4, 53	11
Croatia	0, 79494	5, 41	8, 59	3, 61	27, 75
Lithuania	0, 867599	7, 72	9, 63	5, 6	13, 6
Latvia	0, 826798	6, 12	8, 34	3, 97	15, 8
Slovenia	0, 843189	6, 46	8, 19	4, 56	13, 2
Estonia	0, 811968	5, 37	6, 59	4, 29	10, 4
Cyprus	0, 780476	4, 68	5, 82	3, 65	11, 6
Luxembourg	0, 804808	5, 22	6, 23	3, 89	9, 4
Malta	0, 572356	2, 35	2, 80	1, 92	8,6

Table 3: *Average values of indicators per the EU member state.*

groups are the following: $2 \equiv \{MAL\}$, $3 \equiv \{HUN\}$,
$4 \equiv \{SPA,\ POL,\ POR\}$, $5 \equiv \{ITA,\ ROM,\ GRE,\ AUS,\ CRO,\ EST,\ CYP,\ LUX\ \}$,
$6 \equiv \{GER,\ (UK),\ BUL,\ IRE,\ LAT,\ SLOVEN\}$,
$7 \equiv \{FRA,\ CZE,\ SWE,\ FIN,\ DEN,\ SLOVAK\}$, $8 \equiv \{NET,\ LYT\}$, $9 \equiv \{BEL\}$,
and they are illustrated in Figure 2.

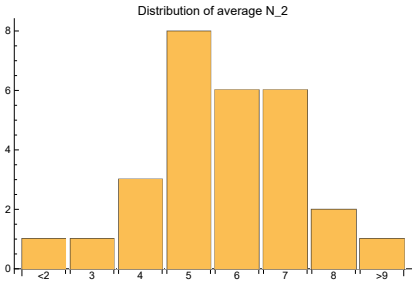


Figure 2: *Average values of the indicator N_2 of each EU member state, rounded and grouped at integer values: 2,3, . . . ,8, ≥ 9.*

With respect to the indicators of the effective number of parties, each EU member state has its specific values of indicators, that also depend on the year in which the elections are held.

2.1. Modified indicators of the number of parties

In order to reduce a sensitivity of indicators N_2, N_1, NP to the formal count n of parties, these indicators are modified by eliminating "tiny" parties. Therefore, parties that are above the threshold of $0,5\%$ (i.e., the set $I' = \{i : \omega_i > 0,005\}$, where $|I'| = n_M$) are taken into account. Then the corresponding modified vote shares are determined by the expression:

$$\omega'_i = \frac{v_i}{\sum_{i\in I'} v_i}\,, \qquad i \in I'\,. \tag{7}$$

Thus, the corresponding modified indicators are obtained as follows. The modified Laakso-Taagepera index has the form (from (2)):

$$MN_2 = \frac{1}{\sum\limits_{i\in I'} \omega_i'^2}\,. \tag{8}$$

In connection with the indicator MN_2, the modified fractionalization can be obtained from (4):

$$MF_2 = 1 - \sum_{i\in I'} (\omega'_i)^2 = 1 - \frac{1}{MN_2}\,. \tag{9}$$

In addition, the modified Wildgen index is defined by the following formula (from (5)):

$$MN_1 = e^{-\sum\limits_{i\in I'} \omega'_i \cdot \ln\omega'_i} \tag{10}$$

The modified Molinar index has the following form (from (6)):

$$MNP = 1 + \frac{\sum_{i\in I'}(\omega'_i)^2 - (\omega'_{[1]})^2}{(\sum_{i\in I'} \omega_i'^2)^2} = 1 + MN_2 - MN_2^2 \cdot (\omega'_{[1]})^2, \tag{11}$$

where $\omega'_{[1]}$ is the largest vote share.

Example 2. *In this example modified indicators MN_2, MN_1 and MNP have been calculated in the cases of elections for the European Parliament in 2004, 2009, 2014, 2019 and 2024 in member states of the European Union (EU), as in Example 1.*

Some results about the indicators and the modified indicators of the effective numbers of parties are given for certain cases of EU member states, in Table 4.

Portugal (2024)	$F_2 = 0,767988$ $MF_2 = 0,762028$	$N_2 = 4,31$ $MN_2 = 4,20$	$N_1 = 5,87$ $MN_1 = 5,46$	$NP = 3,32$ $MNP = 3,26$	n=17 $n_M = 9$
Spain (2024)	$F_2 = 0,770479$ $MF_2 = 0,760097$	$N_2 = 4,35$ $MN_2 = 4,17$	$N_1 = 6,56$ $MN_1 = 5,77$	$NP = 3,10$ $MNP = 3,01$	n=34 $n_M = 11$
Croatia (2024)	$F_2 = 0,791779$ $MF_2 = 0,782055$	$N_2 = 4,80$ $MN_2 = 4,59$	$N_1 = 7,54$ $MN_1 = 6,70$	$NP = 2,96$ $MNP = 2,87$	n=25 $n_M = 14$
Belgium (2024)	$F_2 = 0,903734$ $MF_2 = 0,902692$	$N_2 = 10,39$ $MN_2 = 10,28$	$N_1 = 11,67$ $MN_1 = 11,34$	$NP = 9,03$ $MNP = 8,94$	n=18 $n_M = 13$
Germany (2024)	$F_2 = 0,871574$ $MF_2 = 0,861605$	$N_2 = 7,79$ $MN_2 = 7,23$	$N_1 = 11,06$ $MN_1 = 9,28$	$NP = 5,38$ $MNP = 5,06$	n=35 $n_M = 15$

Table 4: *Comparison of indicators and modified indicators.*

Table 4 shows that the modified indicators have smaller values than the original indicators. The modified indicator MN_1 has got the largest decrease in comparison with the modified indicators MN_2 and MNP.

3. Indicator of the expectation of government stability

After transformation of votes into seats, a government can be formed if it has support of more than 50% of the representatives. The winning coalition is a coalition that has more than 50% of the seats in the parliament. Therefore, the winning coalitions (i.e., a parliamentary majority) are neccessary for government formation in most cases. Greater fractionalization (i.e., greater the effective number of parties) leads to more difficult formation of a parliamentary majority. Political differences between parties makes some winning coalitions almost impossible. There are different approaches within public choice theory regarding the minimal winning coalition [23] and the minimal-connected-winning coalition [2]. For example, the subset of minimal winning coalitions, denoted by C_{dM}, that have political distance which is lower than a given threshold d can be taken into account [4].

With respect to this, one can consider government stability, i.e., how stable a government is or how long it will last. In analysis of government stability one can use the historical approach, or approach based on measuring the expectation for a stable government.

The indicator called the expectation of government stability is defined as follows ([4]):

$$ES = \frac{1}{|C_{dM}|} \sum_{C \in C_{dM}} \frac{w(C)\sigma(C)}{|C|}, \tag{12}$$

where:

C_{dM} is the subset of *minimal winning coalitions* whose *political distance* is lower than a given threshold d,

$\sigma(C)$ is the share of total seats that a minimal winning coalition $C \in C_{dM}$ holds,

$w(C)$ is a weight associated with the coalition $C \in C_{dM}$.

One can see by (12) that the indicator ES has values in $[0, 1]$ and increases when the number of seats $s(C)$ of minimal winning coalition C increases, and when there exist 'large parties' that are frequently contained in the set of C_{dM}. On the other side, indicator ES decreases when the number of coalitions $|C_{dM}|$ increases, and when the number of parties $|C|$ that form a minimal winning coalition C increases.

This indicator is presented by a few cases of elections in certain countries as follows.

Example 3. *In this example indicator ES is calculated in several empirical cases of parliaments after elections in certain countries in the EU. In addition, the indicator ES in the cases of the EU Parliament after elections in 2014 and 2024 is calculated.*

Some obtained results about the values of indicator ES are given in Table 5a, 5b and 5c.

From Table 5a, 5b, 5c, it can be seen that empirical values of this indicator are in accordance with its theoretical properties.

Remark 1. *Notice that if there is only one minimal winning coalition with given threshold, i.e., if* $|C_{dM}| = 1$, *then* $ES = \frac{\sigma(C)}{|C|}$.

3.1. Modified indicator of the expectation of government stability

The indicator ES (12) takes into account the set C_{dM} of minimal winning coalitions with distance threshold d. However, a formally measured political distance between the parties in a minimal winning coalition does not have to significantly affect government stability.

Thus, instead of the set C_{dM}, it can be reasonable to take into account the set of politically feasible winning coalitions, which depends on practical political and social relationships between parties [15]. The set of politically feasible winning coalitions could be determined by political experts (let us denote it by C_P).

Country, Year	Total seats, Quota	C_{dM}	ES
Croatia, 2015	$S = 151$, $q = 76$	$\{\{59, 15, 8\}, \{56, 15, 8\}\}$	$0,089$
Croatia, 2020	$S = 151$, $q = 76$	$\{\{61, 3, 12\}, \{12, 61, 3\}\}$	$0,084$
Croatia, 2024	$S = 151$, $q = 76$	$\{\{61, 12, 4\}, \{10, 42, 5, 11, 4, 4\}\}$	$0,058$
Spain, 2016	$S = 350$, $q = 176$	$\{\{137, 32, 17\}, \{85, 71, 17, 7\}\}$	$0,075$
Slovenia, 2022	$S = 90$, $q = 46$	$\{\{41, 7\}, \{41, 5\}\}$	$0,131$
Austria, 2019	$S = 183$, $q = 92$	$\{\{71, 26\}, \{71, 31\}\}$	$0,136$
Euro. Parl., 2014	$S = 751$, $q = 376$	$\{\{215, 74, 46, 39, 16\}, \{189, 70, 52, 50, 16\}\}$	$0,051$
Euro. Parl., 2024	$S = 720$, $q = 361$	$\{\{188, 77, 136\}, \{188, 78, 84, 25\}\}$	$0,077$

Table 5a: *Values of ES when $|C_{dM}| = 2$.*

Country, Year	Total seats, Quota	C_{dM}	ES
Croatia, 2020	$S = 151$, $q = 76$	$\{\{10, 2, 61, 3\}, \{2, 61, 3, 10\}, \{61, 3, 10, 2\}\}$	$0,042$
Croatia, 2024	$S = 151$, $q = 76$	$\{\{61, 12, 4\}, \{61, 5, 4, 4, 2\}, \{10, 42, 5, 9, 2, 4, 4\}\}$	$0,034$
Slovenia, 2018	$S = 90$, $q = 46$	$\{\{25, 10, 7, 4\}, \{25, 13, 10\}, \{13, 10, 10, 9, 4\}, \{13, 10, 10, 9, 5\}, \{13, 10, 10, 7, 5, 4\}\}$	$0,023$
Austria, 2013	$S = 183$, $q = 92$	$\{\{52, 47\}, \{47, 40, 11\}, \{47, 40, 9\}\}$	$0,067$

Table 5b: *Values of ES when $|C_{dM}| > 2$.*

Country, Year	Total seats, Quota	C_{dM}	ES
Hungary, 2018=2014	$S = 199$, $q = 100$	$\{\{133\}\}$	$0,668$
Austria, 2017	$S = 183$, $q = 92$	$\{\{62, 51\}\}$	$0,309$
Croatia, 2020	$S = 151$, $q = 76$	$\{\{12, 61, 3\}\}$	$0,168$
Croatia, 2024	$S = 151$, $q = 76$	$\{\{61, 12, 4\}\}$	$0,170$

Table 5c: *Values of ES when $|C_{dM}| = 1$.*

In addition, a feasible winning coalition does not have to be a minimal winning coalition. In that case, one can look at the number of parties that are critical in a feasible winning coalition C, denoted by $|C_{crit}|$. (By definition, the party i is critical in the coalition C, if when it exits the coalition C, the coalition $C \setminus \{i\}$ becomes the non-winning.) Here, based on the formula (12), the modified expectation of government stability can be proposed by the following formula:

$$ES_M = \frac{1}{|C_P|} \sum_{C \in C_P} \frac{w(C)\sigma(C)}{|C_{crit}|}, \tag{13}$$

where:

C_P is the set of *feasible winning coalitions*,

$\sigma(C)$ is the share of total seats that a coalition $C \in C_P$ holds,

$w(C)$ is a weight associated with a coalition $C \in C_P$,

$|C_{crit}|$ is the number of parties that are critical in the coalition $C \in C_P$.

Notice that if a feasible winning coalition C is minimal, then it holds $|C_{crit}| = |C|$.
The modified indicator (13) is illustrated by a few examples.

Example 4. *a) The data from results of elections for the Croatian Parliament in 2015 are used. The electoral system in Croatia is considered in [14]. There are $S = 151$ members of Croatian Parliament, so quota $q = 76$ represents the majority votes. Let us assume that there are two feasible winning coalitions, i.e., $C_P = \{\{59, 15, 8, 2, 2, 1\}, \{56, 15, 8, 3, 3, 1, 1, 1\}\}$.*

The calculated value of the modified indicator - the modified expectation of government stability is given in Table 6. In comparison with the corresponding indicator $ES = 0,089$ from Table 5a, one gets $ES_M = 0,1450$.

b) The data from results of elections for the Croatian Parliament in 2024 are used. Assume that there are two feasible winning coalitions, i.e.,
$C_P = \{\{61, 12, 4, 1\}, \{10, 42, 5, 10, 4, 12\}\}$.

The calculated value of the modified indicator - the modified expectation of government stability is given in Table 6. In comparison with the corresponding indicator $ES = 0,058$ *from Table 5a, one gets* $ES_M = 0,076$.

Country, Year	Total seats, Quota	C_P	ES_M
Croatia, 2015	$S = 151$, $q = 76$	$\{\{59, 15, 8, 2, 2, 1\}, \{56, 15, 8, 3, 3, 1, 1, 1\}\}$	0,1450
Croatia, 2024	$S = 151$, $q = 76$	$\{\{61, 12, 4, 1\}, \{10, 42, 5, 10, 4, 12\}\}$	0,076
Spain, 2016	$S = 351$, $q = 176$	$\{\{137, 32, 17, 7, 1\}, \{85, 71, 17, 7, 1\}\}$	0,1049

Table 6: *Values of the modified indicator* ES_M.

c) The Congress of Deputies in Spain after elections in June 2016 is observed. It consists of 350 members. So quota $q = 176$ *represents the majority votes. Let us suppose hypothetical situation, that there are two feasible winning coalitions, i.e.,* $C_P = \{\{137, 32, 17, 7, 1\}, \{85, 71, 17, 7, 1\}\}$.

The corresponding value of the modified indicator - the modified expectation of government stability is given in Table 6. In comparison with the corresponding indicator $ES = 0,075$ *from Table 5a, one gets* $ES_M = 0,1049$.

Remark 2. *In the special case when the government has been formed, where the government coalition* C_G *is winning, but not minimal winning coalition, then* $C_P = \{C_G\}$ *and from* (13) *it follows the formula:*

$$ES_M = \frac{\sigma(C_G)}{|C_{Gcrit}|},$$

where $|C_{Gcrit}|$ *is number of parties in the government coalition* C_G *that are critical.*

Example 5. *a) Given the data for Croatia in 2011:* $S = 151$, $q = 76$, $C_G = \{60, 14, 4, 2\}$, *for the modified expectation of government stability one obtains* $ES_M = \frac{0,530}{2} = 0,265$. *Notice that if the set of minimal winning coalitions is* $C_{dM} = \{\{60, 14, 4\}\}$, *then for the expectation of government stability one obtains* $ES = \frac{0,517}{3} = 0,172$.

b) Given the data for Croatia in 2024: $S = 151$, $q = 76$, $C_G = \{61, 12, 4, 1\}$, *one obtains* $ES_M = \frac{78/151}{3} = 0,172$. *Notice that if the set of minimal winning coalitions is* $C_{dM} = \{\{61, 12, 4\}\}$, *then one obtains* $ES = \frac{77/151}{3} = 0,170$.

4. Conclusion

Electoral systems can be analyzed by means of their many features. With respect to the effective number of parties in a political system, in this paper some known quantitative indicators are considered: the Laakso-Taagepera index (N_2), fractionalization of vote shares (F_2), the Wildgen index (N_1), the Molinar index (NP). These indicators of the effective number of parties are examined from the empirical point of view in the cases of elections for the European Parliament in the EU member states. Each EU member state has its specific values of indicators.

Furthermore, a modification of these indicators is suggested, in order to decrease impact of tiny parties to the values of indicators. This is illustrated by some results about the indicators and the modified indicators of the effective numbers of parties for certain cases of the EU member states. The examples illustrate that the modified indicators have smaller values than the (original) indicators.

With regard to the government stability, in this paper the indicator of the expectation of government stability (ES) is observed. It takes into account the minimal winning coalitions

with distance threshold. Based on several examples of elections in different countries, one can see that empirical values of this indicator are in accordance with its theoretical properties. The question about government stability is too complex. A comparison of mutual values of the expectation of government stability in different cases can show in which cases the expectation of government stability is larger.

In order to extend the definition of this indicator, from the minimal winning coalitions to the politically feasible winning coalitions, a modification of the expectation of government stability is suggested. The modified expectation of government stability is illustrated by some empirical examples of elections. The results in these empirical cases confirm that the modified expectation of government stability (ES_M) is larger than the indicator ES.

With respect to future research, the effective number of parties can be studied as a parameter that changes in successive election cycles. In addition, the number of parliamentary parties can be considered as an indicator that refers to the phase after transformation of votes into seats. With regard to the (modified) expectation of government stability, the variability of this indicator can be studied in empirical cases of successive elections.

Acknowledgements

The authors would like to thank anonymous reviewers and journal editors for their careful reading of the paper and insightful comments that helped us improve the paper.

References

[1] Androniceanu, A., Georgescu, I. and Sabie, O. M. (2022). Comparative research on government effectiveness and political stability in Europe. Administratie si Management Public, 39, 63-76. doi: 10.24818/amp/2022.39-04

[2] Axelrod, R. (1970). Conflict of interest. Chicago, IL: Markham.

[3] Carozzi, F., Cipullo, D. and Repetto, L. (2024). Powers that be? Political alignment, government formation, and government stability. Journal of Public Economics, 230, 105017. doi: 10.1016/j.jpubeco.2023.105017

[4] Cortona, P. G., Manzi, C., Pennisi, A., Ricca, F. and Simeone, B. (1999). Evaluation and Optimization of Electoral Systems. Philadelphia: SIAM.

[5] De Oliveira, A. N. C. (2024). The three constitutional programs against political crises: systems of government and governmental stability in large democracies. Revista Direito GV, São Paulo, v. 20, e2406. doi: 10.1590/2317-6172202406

[6] European Parliament. 2019 election results. Retrieved from: www.europarl.europa.eu (Accessed 2024)

[7] Golosov, G. V. (2010). The Effective Number of Parties: A New Approach. Party Politics, 16(2), 171-192. doi: 10.1177/1354068809339538

[8] Hanretty, C. (2022). Party system polarization and the effective number of parties. Electoral Studies, 76, 102459. doi: 10.1016/j.elect.stud.2022.102459

[9] Herman, V. M. and Taylor, M. (1971). Party systems and government stability. American Political Science Review, 65(1), 28-37. doi: 10.2307/1955041

[10] Holcombe, R. G. (2016). Advanced Introduction to Public Choice. Elgar Advanced Introductions series.

[11] Jelić, S. and Ševerdija, D. (2018). Government Formation Problem. Central European Journal of Operations Research, 26, 659–672. doi: 10.1007/s10100-017-0505-8

[12] Laakso M. and Taagepera, R. (1979). Effective number of parties. Comparative Political Studies, 12(1), 3-27. doi: 10.1177/001041407901200101

[13] Lublin, D. (2024). Extreme events, decentralisation and the effective number of parties. Regional Studies, 1-12. doi: 10.1080/00343404.2024.2311739

[14] Marošević, T., Sabo, K. and Taler, P. (2013). A mathematical model for uniform distribution voters per constituencies. Croatian Operational Research Review, 4(1), 53-64. Retrieved from: hrcak.srce.hr

[15] Marošević, T. and Soldo, I. (2018). Modified indices of political power: a case study of a few parliaments. Central European Journal of Operations Research, 26, 645–657. doi: 10.1007/s10100-017-0487-6
[16] Molinar, J. (1991). Counting the number of parties: An alternative index. American Political Science Review, 85(4), 1383-1391. doi: 10.2307/1963951
[17] Mueller, D. C. (2003). Public Choice (3th ed.). London: Cambridge University Press.
[18] Rae, D. (1967). The Political Consequences of Electoral Laws. New Haven, CT: Yale University Press.
[19] Rae, D., Loosemore, J. and Hanby, V. J. (1973). Thresholds of representation and thresholds of exclusion: an analytical note on electoral systems. Comparative Political Studies, 3(4), 479-488. doi: 10.1177/001041407100300406
[20] Taylor, A. D. and Pacelli, A. M. (2008). Mathematics and Politics. NewYork: Springer.
[21] Taylor, M. (1972). On the Theory of Government Coalition Formation. British Journal of Political Science, 2(3), 361-373. doi: 10.1017/S0007123400008711
[22] Van de Wardt, M. (2017). Explaining the effective number of parties: Beyond the standard model. Electoral Studies, 45, 44-54. doi: 10.1016/j.electstud.2016.11.005
[23] Von Neumann, J. and Morgenstern, O. (1953). Theory of games and economic behavior. Princeton, NJ: Princeton University Press.
[24] Wildgen, J. K. (1971). The measurement of hyperfractionalization. Comparative Political Studies, 4, 233-245.

Croatian Operational Research Review
CRORR **15:2**(2024), 131–144

CB-SEM vs PLS-SEM comparison in estimating the predictors of investment intention

Marija Vuković[1]*

[1] *University of Split, Faculty of Economics, Business and Tourism, Department of Quantitative Methods, Cvite Fiskovića 5, 21000 Split, Croatia*
E-mail: ⟨marija.vukovic@efst.hr⟩

Abstract. This paper compares different structural equation modeling approaches in estimating the predictors of investment intention. Covariance-based structural equation modeling (CB-SEM) and partial least squares structural equation modeling (PLS-SEM) techniques were compared in the estimation of the model according to the theory of planned behavior (TPB). Additionally, the consistent PLS algorithm (PLSc) was taken into consideration in the methods comparison. To determine which factors affect stock investment intention, a TPB model with attitude towards behavior, perceived behavioral control, and subjective norm as independent variables was estimated using three different approaches. The factors in the model were measured using survey indicators and the final sample included 200 Croatian residents. The results mostly show matching conclusions about the investment intention predictors, with a small difference observed in the PLS-SEM method. It can also be concluded that the factor loadings are higher according to PLS-SEM, as well as the indicators of convergent validity and reliability. On the other hand, CB-SEM shows stronger structural paths than PLS-SEM, and PLSc results are closer to those of CB-SEM. While CB-SEM shows better model fit, PLS-SEM shows high predictive power. This research further provides explanations of the differences and guidelines on when to use which approach.

Keywords: CB-SEM, investment intention, PLS-SEM, structural equation modeling

Received: May 14, 2024; accepted: July 5, 2024; available online: October 7, 2024

DOI: 10.17535/crorr.2024.0011

1. Introduction

Structural equation modeling (SEM) is a second-generation multivariate technique aimed at explaining the relationships between multiple variables simultaneously. A specific characteristic of SEM is its ability to incorporate latent variables into the model. Such variables are not directly measurable, but can be indirectly measured with a set of items (indicators) or measured variables [4, 6, 11]. Since latent variables most commonly represent some psychological constructs, their corresponding items are in most cases measured through survey questionnaires.

This specific approach to SEM deals with model estimation based on the differences between the observed and the estimated covariance matrix. Therefore, the "classic" SEM approach is also known as covariance-based SEM (CB-SEM) and it represents a confirmatory method [6, 11, 12]. In order to conduct an analysis using CB-SEM, it is necessary to have an established theoretical background which researchers want to test on their own data. The results of CB-SEM show whether a theory can be confirmed or rejected across different samples [4, 6, 12].

In contrast, the emergence of partial least squares structural equation modeling (PLS-SEM) shows a causal-predictive orientation and a variance-based approach. Thus, the main goal of

*Corresponding author.

PLS-SEM is prediction, obtained by maximizing the explained variance in the dependent variable, making it suitable for theory development and exploratory research [11, 12, 13]. Another method, consistent PLS (PLSc), is a variation of the original PLS and it offers a correction for attenuation to PLS path coefficients. This method usually yields results very similar to those of CB-SEM [6, 12].

The aim of this research is to compare all three methods using the same model with the same dataset. In this way, it is possible to make clearer explanations of the differences and to set some guidelines for choosing the appropriate SEM method. The paper begins with an introduction, followed by a more detailed explanation of each SEM approach. The research context and the model are presented in Section 2. Section 3 describes the data collection and the used methodology, while Section 4 presents the empirical results and a comparison of the studied methods. The conclusion is provided in Section 5.

2. A review of SEM characteristics and research context

2.1. SEM characteristics

According to [11], SEM is a "multivariate technique combining aspects of factor analysis and multiple regression that enables the researcher to simultaneously examine a series of interrelated dependence relationships among the measured variables and latent constructs (variates), as well as between several latent constructs". Some of the main advantages of SEM can be inferred from this definition. Namely, simultaneous estimation of multiple relationships provides a more accurate depiction of the tested theory and accounts for measurement error in the process. However, one of the greatest strengths of SEM lies in its ability to incorporate latent variables into the model, since they cannot be directly measured but are often used in behavioral and social research. Instead, these variables are represented by a set of indicators obtained through data collection, commonly through primary data [6, 11, 12, 22, 24, 27, 32]. SEM can be divided into two different approaches: CB-SEM and PLS-SEM.

CB-SEM is a confirmatory method based on the differences between the observed and estimated covariance matrices. This allows for testing of the model fit, i.e. the extent to which the theoretically proposed model is confirmed by empirical data [4, 6, 11, 22]. The modeling can be conducted in one or two steps. Two-step modeling refers to the process where confirmatory factor analysis (CFA) is used in the first step to obtain model fit and construct validity measures for the measurement model. Only after achieving an acceptable fit, the structural model is tested for path significance. In contrast, one-step modeling tests the overall model fit without separating measurement and structural models [6, 11]. However, in both cases, model fit is obtained, as well as the assessment of construct validity. A valid result from one-step modeling implies that two-step modeling would also yield satisfactory results.

Overall model fit is primarily evaluated with the chi-square (χ^2) test, the sole inferential statistic which compares the statistical difference between the observed and estimated covariance matrices. A good fit implies that the differences are statistically insignificant, making chi-square a measure of badness of fit [6, 11, 15]. However, chi-square has its limitations in the fact that it assumes multivariate normality of the data, so large deviations from this assumption could result in a significant result even if the model is correctly specified. Additionally, chi-square is sensitive to sample size, tending to increase with it, consequently leading to a significant result [11, 15, 16]. Due to these limitations, other model fit measures are recommended for reporting in CB-SEM analysis. One of these measures is the normed chi-square (χ^2/df), where a ratio between 2 and 5 is considered acceptable [6, 15, 16]. Other measures include the root mean square error of approximation (RMSEA) and standardized root mean residual (SRMR), which require lower values in order to establish good model fit (RMSEA$<$0.08 or $<$0.10; SRMR$<$0.05). Additionally, goodness-of-fit measures, such as comparative fit index

(CFI), Tucker Lewis index (TLI), goodness-of-fit index (GFI), adjusted goodness-of-fit index (AGFI), normed fit index (NFI) compare the model fit against a null or independent model. Therefore, higher values, usually above 0.90 are required for a good model fit [6, 11, 15, 16].

PLS-SEM is a causal-predictive approach aimed at maximizing the explained variance in dependent latent constructs, hence it is also known as a variance-based approach to SEM. In addition, it also evaluates data quality obtained through measurement model analysis [6, 12, 26]. In PLS-SEM, the assumption of multivariate normality of data is not required to estimate the model, distinguishing it as a non-parametric approach to SEM. While CB-SEM requires large samples, PLS-SEM can effectively handle both larger and smaller samples. One of its advantages is also the ability to handle single-item constructs, which is problematic for CB-SEM because of model identification issues. Furthermore, PLS-SEM offers greater statistical power, making it suitable for theory development in exploratory research [4, 10, 11, 12, 22, 26]. In social sciences, most of the measurement is reflective (the construct causes the indicators), but sometimes formative measurement can occur (the indicators cause the construct). While the indicators in the reflective measurement are interchangeable, it is not the case in the formative measurement [12]. In case of formative models, PLS-SEM should be used rather than CB-SEM. The treatment of latent constructs should also be considered, since the constructs can be viewed as common factors or composites. The common factor approach is widely used in CB-SEM and it is based solely on common variance in the data, while the composite approach, which includes common, specific, and error variance, is used in PLS-SEM [10, 12, 22]. Common factor assumes causal indicators, which should fully measure a certain concept along with an error term. Composite indicators are considered an approximation of a concept, which is a more realistic view in social sciences. According to [10], due to random error in composite models and factor indeterminacy in common factor models, both approaches yield only approximations of theoretical concepts. In other words, "common factor proxies cannot be assumed to carry greater significance than composite proxies in regard to the existence or nature of conceptual variables" [10].

Both CB-SEM and PLS-SEM require the analysis of the measurement and structural models (in PLS-SEM: the outer and inner model). The first part of the analysis verifies the validity and reliability of the constructs, while the second part deals with the assessment of structural paths and their significance. In contrast to CB-SEM, which is a parametric approach that yields statistical significance as a test result, PLS-SEM, as a non-parametric method, relies on the bootstrapping procedure in order to obtain significance [10, 11, 12, 22]. As previously mentioned, CB-SEM is focused on model fit and theory testing, whereas PLS-SEM is prediction oriented. Therefore, even though some model fit measures can be obtained for PLS-SEM, the main focus lies in analyzing the in-sample and out-of-sample predictive power of the model. In-sample predictive power is assessed with the coefficient of determination (R^2). PLS predict procedure is recommended for assessing the model's out-of-sample predictive power [29]. Lately, the cross-validated predictive ability test (CVPAT) has also been developed and recommended as a prediction-oriented tool [28].

In an attempt to provide a factor-based approach within PLS-SEM, the consistent PLS-SEM approach (PLSc) was proposed. It uses Nunnally and Bernstein's equation as a correction factor to the traditional PLS algorithm, obtaining consistent construct correlations and indicator loadings if the common factor model holds true [6, 7, 9, 12]. PLSc has been shown to yield results very similar to those of CB-SEM, while retaining the advantages of PLS-SEM, but its statistical power is somewhat lower [7, 9, 12, 34].

Considering all of the above, it is clear that CB-SEM and PLS-SEM should be viewed as complementary rather than competitive. The choice of method should be based on the research goal (purely confirmatory or causal-predictive), the measurement philosophy, the sample size, and data characteristics.

2.2. Research context

The focus of the paper is to compare different SEM approaches for model estimation. The model used in the research is the model of the theory of planned behavior (TPB) in the context of stock market investment intention, as an illustrative example. Therefore, the model only serves as an example in the methodological comparison. The concept and essence of the model are briefly described to give context to the reader. TPB assumes that different motivational factors, such as attitude towards behavior, subjective norm, and perceived behavioral control, significantly influence a certain behavioral intention [2, 3, 20, 23, 33]. Intention is the central factor in TPB, capturing these motivational factors and indicating the level of effort people are willing to put into performing a certain behavior [2]. In this paper, the intention is specifically oriented towards stock market investments.

Briefly, attitude towards behavior can be defined as the extent to which a person considers a behavior pleasant or unpleasant [2, 3, 20]. Therefore, a favorable and positive attitude leads to a higher intention to perform a certain behavior [2, 20, 33]. When considering stock market investment intention, this assumption was found to be true in previous research [3, 19, 20, 23]. Therefore, the first research hypothesis is proposed:

H1. Attitude towards behavior significantly affects investment intention.

Subjective norm represents the perceived social pressure whether to perform a behavior or not. People will often behave in a certain way if they are under a higher social pressure. In other words, if they perceive that people who are close to them and who influence them in daily life think that they should perform a certain action, they might do it under pressure, even if they do not want to [2, 3, 20]. Most people are often concerned about the opinions of others, thus subjective norm might often be the most influential factor affecting behavioral intention. It was previously found that subjective norm positively affects individual's stock investment intention, showing the importance of social pressure in forming a higher intention towards a certain behavior [3, 19]. However, it was also found that among younger generations (Y and Z), subjective norm was not an important predictor of investment intention or it even had a negative impact [20, 23]. Clearly, younger people have less role models who might encourage them for investing, or they are generally less prone to be under social pressure, since they are less affected by the opinions of others. Nevertheless, social influence still remains one of the most important factors in predicting certain behavior, leading to the formulation of the second research hypothesis:

H2. Subjective norm significantly affects investment intention.

Lastly, perceived behavioral control refers to the perception of ease or difficulty in performing a specific behavior. It reflects past experiences and expected obstacles [2, 20, 23]. It is expected that people with higher perceived behavioral control will consequently have higher intentions towards performing a behavior. This relationship has been confirmed in most of the previous research, showing that people with stronger control show higher stock investment intentions [3, 19, 20]. According to these findings, the third hypothesis is proposed:

H3. Perceived behavioral control significantly affects investment intention.

The research model investigates how attitude towards behavior, subjective norm, and perceived behavioral control affect investment intention in Croatia. There is no similar research in Croatia and other emerging markets, which have not yet been studied much in the context of behavioral finance. Croatians generally exhibit low engagement in stock market activities, they possess lower financial literacy and prefer to invest in real estate [31]. However, as TPB proposes general factors in prediction of a certain behavior, it is hypothesized that the same general pattern should be applicable to any sample, including Croatian.

3. Data and methodology

3.1. Research instrument and data collection

Data were collected via a survey questionnaire, distributed to the general Croatian population online from May to July 2023. The first part of the survey consists of questions regarding the socio-demographic traits of the respondents, while the second part is related to the factors of the theory of planned behavior. Namely, the questions were represented as statements measuring the attitude towards behavior (ATT), perceived behavioral control (PBC), subjective norm (SN), and investment intention (INT) on a 5-point Likert scale (1=strongly disagree, 5=strongly agree). The statements from the survey were based on the research of [23] and they represent the items for further structural modeling. The items are shown in Table 1.

	Attitude towards behavior (ATT)
ATT1	I think that investing in the stock market can enhance the financial knowledge of individuals.
ATT2	I think that stock investment is meaningful.
ATT3	I think that stock investment is a good idea.
	Perceived behavioral control (PBC)
PBC1	I have enough time for stock investment.
PBC2	I have enough money for stock investment.
	Subjective norm (SN)
SN1	I will participate in stock investment if my spouse thinks it is useful.
SN2	I will participate in stock investment if my family approves it.
SN3	I will participate in stock investment if my colleagues do.
SN4	I will participate in stock investment if I have proven friend success on it.
	Investment intention (INT)
INT1	I intend to engage in stock investment in the near future.
INT2	I will recommend others to invest in the stock market.
INT3	I will continue to invest in the stock market.

Table 1: *Items in the model.*

3.2. Data

As previously mentioned, this research focuses on the Croatian general population. Data were collected online via e-mail, social networks, etc. Since there is no "one-size-fits-all" formula for sample size in SEM, there are only general guidelines. According to [11], the minimum sample size for models with seven or fewer constructs and modest communalities with values around 0.5 is 150 cases. The research model has four constructs, and the communalities range from 0.49 to 0.814. However, the sample should be increased in case of deviations from multivariate normality, as is the case in this research. A priori sample size was determined with Daniel Soper's formula [30], which takes into account the number of indicators and constructs in the model, the anticipated effect size, and the desired probability and statistical power levels. With 4 constructs and 12 indicators, a medium anticipated effect size (0.3), a 5% probability level, and an 80% desired statistical power level, the recommended minimum sample size was 200. The calculator was used in order to get the suggested sample size as accurately as possible. Simpler models require smaller samples, and a median sample size, often generally recommended as a minimum, is 200 cases [11, 18]. The "10 times rule" can also be considered as a rough guideline for PLS-SEM, stating that the sample should be 10 times the number of arrows pointing at a construct, which is three in this research, implying a sample of 30 cases is needed [11]. These rules of thumb are being abandoned in favor of methods that take statistical power into account. Thus, according to [12]'s sample size recommendation in PLS-SEM, for a statistical power of 80%, with a maximum number of three independent latent variables, as in this model, a sample size of only 37 cases would be needed to achieve a statistical power of 80% for detecting R^2

values of at least 0.25, with a 5% probability of error. The final sample consists of 200 Croatian residents, which is far beyond the proposed limit.

When looking at gender, most of the respondents are female (61.5%) compared to male (38.5%). Most of the respondents are 18-25 years old (24.5%), followed by those aged 36-45 and 46-55 with the same share of 22.5%. 13.5% of people are 26-35, and 12.5% are 56-65 years old. Only 4.5% of the respondents are 66 or older. The age structure is obviously related to the sample structure according to work experience, since most of the respondents have less than 5 years of work experience (30%). The second largest group represents people with more than 20 years of work experience (23.5%). 21.5% of the respondents have 5-10 years of work experience, while 11.5% of the respondents have worked 11-15 years, and 13.5% of them have worked for 16-20 years. The main income source for the respondents is their monthly salary (71.5%), pension is the source for 5.5% of the people, social care for 1%, and the rest of the respondents (22%) have stated that they have other main income source. Most of the people are highly educated, since only 28% have not finished any level of college education, and the remaining 72% have obtained at least bachelor's degree or a higher level of education.

3.3. Methodology

Data analysis was conducted with SmartPLS 4 software [25], which was initially developed for PLS-SEM estimation, but currently has the ability to estimate both PLS-SEM and CB-SEM models. Before modeling, the multivariate normality of the data was examined with Mardia's multivariate skewness and kurtosis tests, which showed that the data deviate from this assumption ($p<0.001$) [21]. Since the items were measured on a Likert scale, this is not a surprising result. Modeling with maximum likelihood (ML) estimation was continued despite this result, which goes against the requirements of CB-SEM. Estimation with the weighted least squares mean and variance adjusted (WLSMV) estimator for ordinal data was done as an additional control, and it was found that the results do not significantly differ compared to the ML estimator. Moreover, the research model is quite simple and the sample size is sufficiently large, thus contributing to more reliable estimates [18]. This also implies that the estimates are robust to data non-normality and that the model is well specified, since it relies on TPB. Namely, TPB is a widely established theory which can be used for confirmatory purposes, so the default ML estimation is used to compare the different approaches, because it is most common and it represents the essence of CB-SEM most precisely. Additionally, as [5] proposed in his comparison of PLS-SEM with CB-SEM's different estimators, the CB-SEM results among different estimators were not contradictory.

In the first step, a CFA model was estimated as a prerequisite for further structural modeling, as suggested by [11]. After establishing a good model fit, CFA was followed by structural model analysis with the CB-SEM approach. The first SEM analysis refers to the measurement model. Model fit measures were examined, followed by the standard analysis of convergent validity using average variance extracted (AVE) and reliability using Cronbach's alpha and composite reliability [6, 11]. Two methods were used to assess the discriminant validity: Fornell and Larcker criterion and heterotrait-monotrait (HTMT) ratio of correlations. According to the Fornell and Larcker criterion, the correlations between the constructs are compared with the square root of the AVE of each construct. It is assumed that a construct should be better at explaining its own indicators' variance rather than the indicators of other constructs [1, 8, 11, 14]. However, this approach has some flaws, because it tends to overstate discriminant validity problems, especially if indicator loadings vary strongly [1, 8]. Thus, recent research has shown that heterotrait-monotrait (HTMT) ratio of correlations is a better choice for discriminant validity testing. HTMT measures "the ratio of the between-trait correlations to the within-trait correlations" [12], i.e. it compares the correlations of the indicators measuring different constructs to the correlations of indicators measuring the same construct. Therefore, lower

HTMT values are preferable. The threshold is 0.85 or a more flexible value of 0.90. HTMT inference can also be calculated to further demonstrate discriminant validity [12, 14]. This measurement model analysis in the PLS-SEM approach is called the outer model analysis, but it includes the same steps as in CB-SEM for reflective models [12]. Therefore, the results for validity and reliability testing are first presented for all three techniques, according to the aforementioned guidelines.

Afterwards, the structural model (or in PLS-SEM: the inner model) is assessed. According to [11] and [12], this part of the analysis explores the strength, direction and significance of the structural paths. CB-SEM gives p-values as a standard output of the test, while for PLS-SEM, the bootstrapping procedure is used to obtain the significance of the paths [10, 11, 12, 22]. Lastly, only for the PLS-SEM approach, predictive measures were also analyzed with the coefficient of determination (R^2) and the PLSpredict procedure [12, 29].

4. Empirical results and discussion

4.1. Validity and reliability analysis

The first step of the analysis prior to CB-SEM is CFA. Before conducting this analysis, indicator multicollinearity was assessed with variance inflation factor (VIF) values. All values are below the threshold of 5, indicating that there is no multicollinearity problem (Table 2). It obtained satisfactory model fit, except for the significant chi-square result, most likely caused by data non-normality and the sample size (χ^2=129.937, p<0.001; χ^2/df=2.707; RMSEA=0.092; SRMR=0.043; GFI=0.904; NFI=0.918; TLI=0.926; CFI=0.946). The second step is to estimate the structural model, which was first done for the CB-SEM approach. The results show that the model fit measures are exactly the same as for the CFA, i.e. a significant chi-square, but satisfactory values for all other measures (χ^2=129.937, p<0.001; χ^2/df= 2.707; RMSEA=0.092; SRMR=0.043; GFI=0.904; NFI=0.918; TLI=0.926; CFI=0.946). PLS-SEM and PLSc were also estimated. Even though their main focus is not on model fit, and these measures in the context of PLS-SEM are still evolving and should be taken with caution, SRMR and NFI are available in the output for comparison with CB-SEM. PLS-SEM model fit was not satisfactory (SRMR=0.066, NFI=0.805), while it was good for PLSc (SRMR= 0.048, NFI= 0.907). PLSc results are close to those of CB-SEM. Measurement (outer) model evaluation was done through validity and reliability analysis. The results for all techniques can be seen in Table 2.

	CB-SEM			PLS-SEM			Consistent PLS-SEM				
	Loading	AVE	Composite reliability	Loading	AVE	Composite reliability	Loading	AVE	Composite reliability	Cronbach's alpha	VIF
Attitude toward behavior (ATT)											
ATT1	0.692	0.632	0.839	0.811	0.749	0.899	0.713	0.628	0.835	0.832	1.632
ATT2	0.851			0.893			0.828				2.253
ATT3	0.833			0.889			0.831				2.206
Perceived behavioral control (PBC)											
PBC1	0.812	0.633	0.775	0.912	0.816	0.899	0.830	0.635	0.776	0.775	1.668
PBC2	0.779			0.895			0.762				1.668
Subjective norm (SN)											
SN1	0.585	0.524	0.817	0.702	0.636	0.874	0.598	0.523	0.812	0.809	1.476
SN2	0.701			0.785			0.659				1.723
SN3	0.814			0.844			0.768				2.138
SN4	0.773			0.850			0.843				2.060
Investment intention (INT)											
INT1	0.918	0.769	0.909	0.941	0.842	0.941	0.885	0.764	0.906	0.906	3.860
INT2	0.864			0.910			0.887				2.849
INT3	0.847			0.902			0.849				2.769

Table 2: *Convergent validity and reliability for the constructs.*

Indicator loadings are substantially high (>0.7) for all constructs, with a few exceptions for CB-SEM and PLSc. However, these small deviations did not affect validity. Namely, all AVE values exceed the cutoff value of 0.5, showing that each construct explains more than 50% of the variance in its indicators, thus confirming the convergent validity of the constructs across all techniques [11, 12]. Cronbach's alpha and composite reliability were both above the threshold of 0.7, demonstrating the reliability of the constructs. These results further confirm that using only two indicators to represent the construct PBC is valid. Even though SEM often requires at least three indicators per construct, there is no magic number of indicators per construct. More indicators make the model more complex, and sometimes it is better to leave some items out if they are highly redundant [18]. From an empirical point of view, these two indicators measuring PBC represent the construct very well according to all aforementioned measures of validity and reliability. Furthermore, the model has no identification issues despite this lower number of indicators for PBC, since it also includes other variables. Namely, as [17] suggest, the model should include at least two correlated latent constructs and two indicators per construct, which correlate with an indicator of another construct, but the indicators' errors should be uncorrelated [11, 17, 18]. This is true for the research model, implying that there is a sufficient number of indicators per construct [17]. From a theoretical point of view, PBC can be understood as the perceived ease or difficulty in performing a certain behavior [2]. Thus, in the context of stock investment intention, time and money can be considered as crucial factors, since time availability and sufficient financial resources are essential for stock investment engagement. These factors inevitably influence investor's confidence and ability to invest, which accurately reflects PBC. Additionally, the goal was to keep the model parsimonious for easier estimation, and to compare the performance of different SEM approaches under non-perfect conditions.

Discriminant validity was tested with both Fornell and Larcker criterion and the HTMT ratio. The results based on Fornell and Larcker criterion for all models separately are shown in Table 3, while HTMT is the same across all methods and is shown in Table 4. When observing discriminant validity through the Fornell and Larcker approach, it can be seen that there is a potential problem with ATT and INT, and SN is highly correlated with all other constructs in the CB-SEM model and PLSc. The values are very similar. In contrast, in PLS-SEM, all of the AVE square root values are higher than the correlations with other constructs, fully supporting discriminant validity [1, 8, 12]. The reason for this can be found in the fact that this descriptive approach to discriminant validity tends to overstate the problem, even more when there are higher variations of the indicator loadings. This is exactly the case in CB-SEM and PLSc models, where there are higher variations of the loading values for ATT and SN, while in the PLS-SEM the variations of the loadings are not so high. In order to overcome this issue, discriminant validity was also assessed with the HTMT ratio, showing that all values are below the threshold of 0.9, confirming discriminant validity. Since the highest correlation was found between ATT and INT, HTMT bootstrap 95% confidence intervals were calculated for additional verification. Discriminant validity was additionally supported, since the confidence interval does not span the value of 1 (0.822-0.957) [12, 14].

	CB-SEM				PLS-SEM				Consistent PLS-SEM			
	ATT	INT	PBC	SN	ATT	INT	PBC	SN	ATT	INT	PBC	SN
ATT	0.795				0.865				0.793			
INT	0.871	0.877			0.776	0.918			0.890	0.874		
PBC	0.767	0.789	0.796		0.607	0.662	0.903		0.751	0.788	0.797	
SN	0.758	0.636	0.767	0.724	0.648	0.556	0.609	0.798	0.778	0.642	0.760	0.723

Table 3: *Discriminant validity according to Fornell-Larcker criterion.*

	ATT	INT	PBC	SN
ATT				
INT	0.892			
PBC	0.748	0.788		
SN	0.793	0.642	0.767	

Table 4: *Discriminant validity according to HTMT ratio.*

4.2. Structural model analysis

Structural model results based on each technique separately are shown in Table 5, and the path diagrams are seen in Figures 1-3. It can be concluded that, according to all models, ATT has a positive and significant effect on INT. This is expected, since a more positive attitude towards investing implies that a person will actually show a higher intention to invest. PBC also positively and significantly affects INT, regardless of the method used. This shows that people with a higher perception of control, in terms of time and money, will consequently have higher investment intentions. Lastly, SN has a negative effect on INT in all of the models. However, its strength is significantly lower for PLS-SEM, while it is highest in PLSc, though relatively close to the coefficient in CB-SEM. Since there is practically no effect (β=-0.025) in PLS-SEM, the path is not significant. In CB-SEM and PLSc, this path is significant at the 0.10 significance level. It implies that people under higher social pressure to invest will have lower investment intentions, which is an unexpected result. This can be explained by the fact that the sample consists mostly of younger people, who care less about social influence and they might find other people unreliable, thus choosing not to follow their advice.

In general, all path coefficients are lowest for PLS-SEM and highest for PLSc. Effect sizes are also presented in Table 5, to determine whether each exogenous construct has a substantive impact on the endogenous construct (INT). For CB-SEM and PLSc, the results are similar, with PLSc yielding somewhat higher values. ATT has the highest effect on INT in all observed models, and its effect is considered large. In CB-SEM, PBC has a medium effect size, and SN has a small effect size. These effects become strong for PBC, and medium for SN in the PLSc model. In contrast, PLS-SEM effect sizes are significantly lower. ATT shows a large effect on INT, PBC has a medium effect, and SN practically has no effect on INT in the PLS-SEM model.

According to the results from CB-SEM and PLSc, all research hypotheses can be accepted as true, i.e. all factors of TPB significantly affect investment intention. On the other hand, according to PLS-SEM results, only H1 and H3 can be accepted, while H2 should be rejected, since SN has no significant impact on INT.

Path	CB-SEM			PLS-SEM			Consistent PLS-SEM		
	Path coefficient	p-value	Effect size	Path coefficient	p-value	Effect size	Path coefficient	p-value	Effect size
ATT → INT	0.743	<0.001	1.043	0.604	<0.001	0.546	0.828	<0.001	1.572
PBC → INT	0.397	0.002	0.266	0.311	<0.001	0.157	0.397	0.014	0.387
SN → INT	-0.232	0.054	0.101	-0.025	0.705	0.001	-0.304	0.068	0.205

Table 5: *Structural model coefficients.*

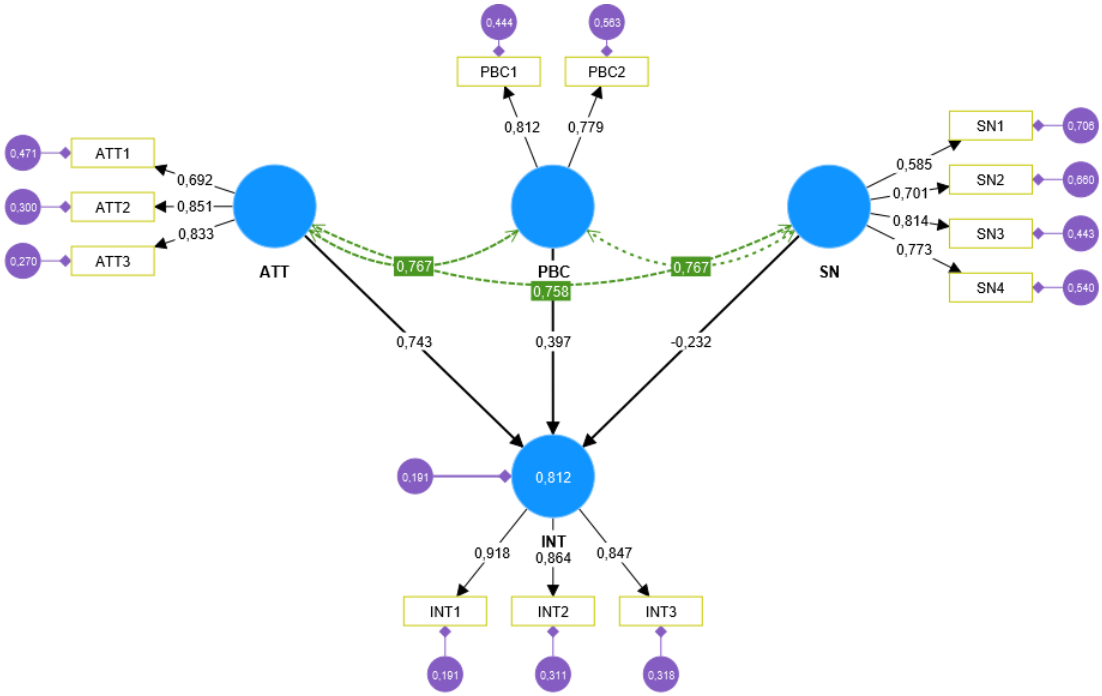


Figure 1: *Path diagram with standardized estimates (CB-SEM).*

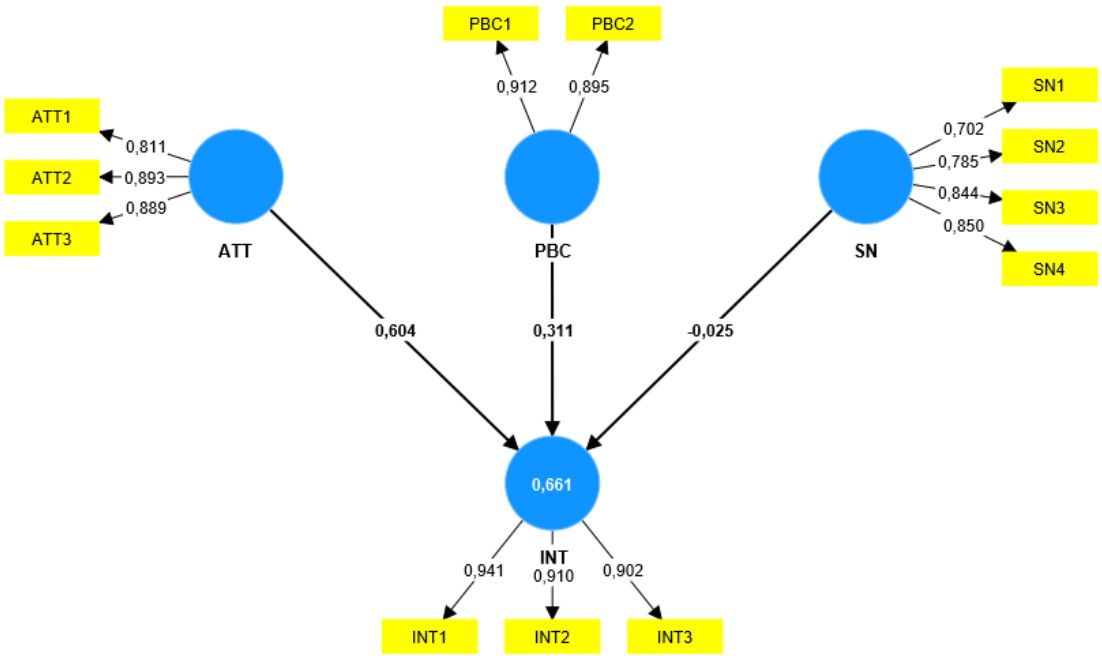


Figure 2: *Path diagram with standardized estimates (PLS-SEM).*

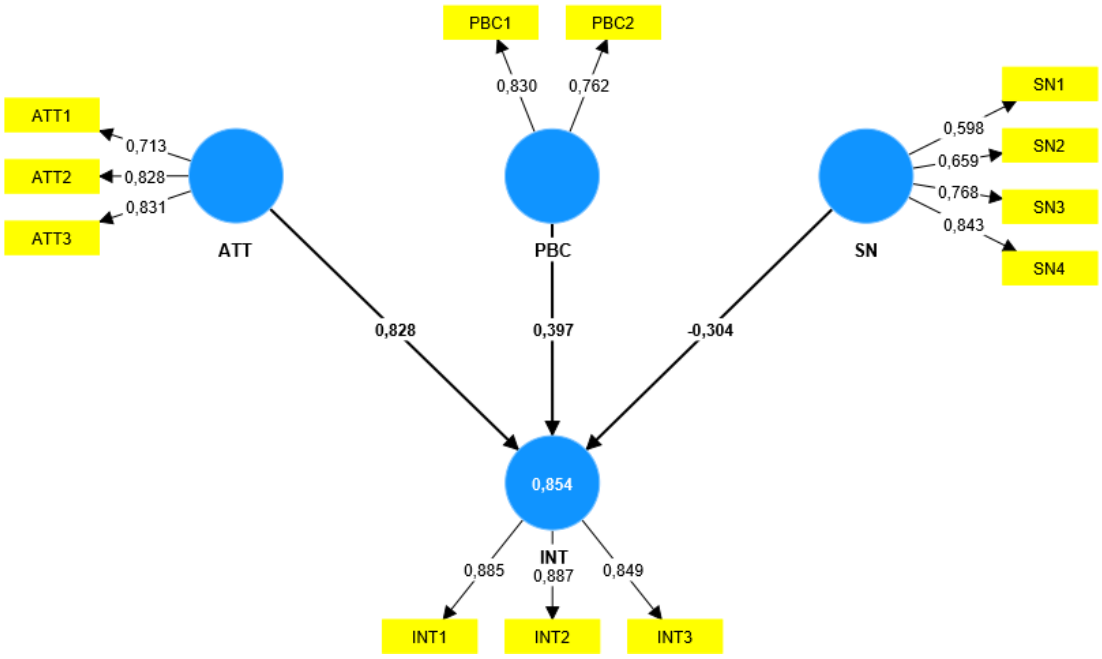


Figure 3: *Path diagram with standardized estimates (PLSc).*

All path diagrams show the R^2 for INT, which is very high for all models. PLS-SEM has the lowest value (0.661), while the values are higher and very similar for CB-SEM (0.812) and PLSc (0.854). This shows that the combination of all exogenous constructs in the models explains a very high proportion of the variance in INT. Thus, all models have high explanatory power.

	Q^2predict	PLS-SEM_RMSE	PLS-SEM_MAE	LM_RMSE	LM_MAE
INT1	0.555	0.738	0.539	0.760	0.560
INT2	0.562	0.738	0.538	0.751	0.547
INT3	0.511	0.744	0.558	0.746	0.565

Table 6: *PLSpredict results.*

PLS-SEM focuses on prediction, so out-of-sample predictive power was tested for this technique with the PLSpredict procedure. According to [29], in this procedure the focus should be on the indicators of the key endogenous construct. This model has only one endogenous construct (INT), so prediction errors of its indicators were analyzed. The results show that all root mean squared error (RMSE) and mean absolute error (MAE) values are lower in the PLS model compared to the LM model, confirming the high predictive power of the model [29]. This is further confirmed with CVPAT, since PLS predictions significantly outperformed the prediction benchmarks ($p<0.001$) [28].

5. Conclusion

This research focuses on comparing different SEM approaches (CB-SEM, PLS-SEM, and PLSc) on a model of the TPB in the context of investment intentions. It can be concluded that the evaluation of the measurement model does not differ significantly among different methods. Indicator loadings, validity, and reliability measures are somewhat higher when estimating a PLS-SEM model. CB-SEM and PLSc results are very similar. The same conclusion can be

made for structural modeling, where CB-SEM and PLSc path coefficients are very close, while PLS-SEM structural paths are weaker. Measures of model fit are satisfactory for CB-SEM, confirming that the theoretical model holds true for the sample. PLSc output provides only a few measures, which are also almost the same as in CB-SEM. On the other hand, PLS-SEM's model fit measures are slightly worse, but this is not essential, since it is not a confirmatory technique. PLS-SEM should rather focus on the R^2 for the model's in-sample predictive power and other tests (PLSpredict, CVPAT) for out-of-sample predictive power. All of the tests show that the model has high predictive abilities and can be reliable for replication with new data. These findings are in line with previous research dealing with SEM estimator comparison [4, 6, 34] on other theoretical models.

Researchers and practitioners should note that the appropriate SEM approach should be chosen according to the research goal as the most important criterion. For purely confirmatory purposes, one should choose CB-SEM, and for predictive purposes, PLS-SEM should be chosen. PLSc should be used instead of CB-SEM if some of the assumptions (normality, sufficient sample size) are not met, but the goal is still confirmatory. Thus, PLS-SEM cannot be chosen over CB-SEM solely due to a small sample or a violation of data normality, although these could be some of the reasons for the preference of PLS-SEM. However, these are not the most important aspects of the method choice. PLS-SEM has recently gained more popularity because it is causal-predictive, and it is assumed that this is the case in most social research. Therefore, PLS-SEM should be chosen in theory development research. Nevertheless, CB-SEM should not be overlooked in cases of confirmatory analysis. It is not excluded to use both methods complementary and comparatively.

Although this paper shows the comparison of three different approaches to SEM, a very simple model was used, which can be considered a limitation. More complex models can be compared across these methods to draw further conclusions. Even though this model provided a proper solution despite the lower number of indicators for PBC construct, researchers are advised to always ensure a larger number of potential indicators in their research, since some of them can be left out in order to establish construct validity. Researchers are encouraged to use at least three indicators per construct whenever possible to avoid potential estimation issues. Future research can also be based on the comparison of these SEM approaches for a unique model under different conditions. For example, a model can be estimated for smaller and larger sample sizes to see if there are any deviations across the three different SEM approaches depending on the sample size.

References

[1] Afthanorhan, A., Ghazali, P. L. and Rashid, N. (2021). Discriminant validity: A comparison of CBSEM and consistent PLS using Fornell & Larcker and HTMT approaches. Journal of Physics: Conference Series, 1874(1), 012085. doi: 10.1088/1742-6596/1874/1/012085

[2] Ajzen, I. (1991). The theory of planned behavior. Organizational behavior and human decision processes, 50(2), 179-211. doi: 10.1016/0749-5978(91)90020-T

[3] Akhtar, F. and Das, N. (2019). Predictors of investment intention in Indian stock markets: Extending the theory of planned behavior. International Journal of Bank Marketing, 37(1), 97-119. doi: 10.1108/IJBM-08-2017-0167

[4] Astrachan, C. B., Patel, V. K. and Wanzenried, G. (2014). A comparative study of CB-SEM and PLS-SEM for theory development in family firm research. Journal of family business strategy, 5(1), 116-128. doi: 10.1016/j.jfbs.2013.12.002

[5] Civelek, M. E. (2018). Comparison of covariance-based and partial least square structural equation modeling methods under non-normal distribution and small sample size limitations. Eurasian Academy of Sciences Eurasian Econometrics, Statistics & Emiprical Economics Journal, 10, 39-50. Retrieved from: https://ssrn.com/abstract=3332989

[6] Dash, G. and Paul, J. (2021). CB-SEM vs PLS-SEM methods for research in social sciences and technology forecasting. Technological Forecasting and Social Change, 173, 121092. doi: 10.1016/j.techfore.2021.121092
[7] Dijkstra, T. K. and Henseler, J. (2015). Consistent partial least squares path modeling. MIS quarterly, 39(2), 297-316. doi: 10.25300/MISQ/2015/39.2.02
[8] Fauzi, M. A. (2022). Partial least square structural equation modelling (PLS-SEM) in knowledge management studies: Knowledge sharing in virtual communities. Knowledge Management and E-Learning, 14(1), 103-124. doi: 10.34105/j.kmel.2022.14.007
[9] Garson, G. D. (2016). Partial Least Squares: Regression and Structural Equation Models. Asheboro, NC: Statistical Associates Publishers.
[10] Hair, J. F., Matthews, L. M., Matthews, R. L. and Sarstedt, M. (2017a). PLS-SEM or CB-SEM: updated guidelines on which method to use. International Journal of Multivariate Data Analysis, 1(2), 107-123. doi: 10.1504/IJMDA.2017.087624
[11] Hair, J. F., Black, W. C., Babin, B. J. and Anderson, R. E. (2019). Multivariate Data Analysis. 8th Ed. Andover: Cengage Learning.
[12] Hair, J. F., Hult, G. T. M., Ringle, C. M. and Sarstedt, M. (2017b). A Primer on Partial Least Squares Structural Equation Modeling (PLS-SEM). 2nd Edition. Los Angeles: Sage Publications.
[13] Henseler, J. (2018). Partial least squares path modeling: Quo vadis?. Quality & Quantity, 52 (1), 1-8. doi: 10.1007/s11135-018-0689-6
[14] Henseler, J., Ringle, C. M. and Sarstedt, M. (2015). A New Criterion for Assessing Discriminant Validity in Variance-based Structural Equation Modeling. Journal of the Academy of Marketing Science, 43(1), 115-135. doi: 10.1007/s11747-014-0403-8
[15] Hooper, D., Coughlan, J. and Mullen, M. R. (2008). Structural Equation Modelling: Guidelines for Determining Model Fit. The Electronic Journal of Business Research Methods, 6(1), 53-60. doi: 10.21427/D7CF7R
[16] Iacobucci, D. (2010). Structural equations modeling: Fit indices, sample size, and advanced topics. Journal of consumer psychology, 20(1), 90-98. doi: 10.1016/j.jcps.2009.09.003
[17] Kenny, D. A. and Milan, S. (2012). Identification: A nontechnical discussion of a technical issue. In: Hoyle, R. H. (Ed). Handbook of structural equation modeling. New York: The Guilford Press, 145-163.
[18] Kline, R. B. (2023). Principles and practice of structural equation modeling. 5th Edition. New York: The Guilford Press.
[19] Mahardhika, A. S. and Zakiyah, T. (2020). Millennials' intention in stock investment: extended theory of planned behavior. Riset Akuntansi dan Keuangan Indonesia, 5(1), 83-91. doi: 10.23917/reaksi.v5i1.10268
[20] Malzara, V. R. B., Widyastuti, U. and Buchdadi, A. D. (2023). Analysis of Gen Z's Green Investment Intention: The Application of Theory of Planned Behavior. Jurnal Dinamika Manajemen Dan Bisnis, 6(2), 63-84. doi: 10.21009/JDMB.06.2.5
[21] Mardia, K. V. (1970). Measures of multivariate skewness and kurtosis with applications. Biometrika, 57(3), 519-530. doi: 10.2307/2334770
[22] Mohamad, M., Afthanorhan, A., Awang, Z. and Mohammad, M. (2019). Comparison between CB-SEM and PLS-SEM: Testing and confirming the maqasid syariah quality of life measurement model. The Journal of Social Sciences Research, 5(3), 608-614. doi: 10.32861/JSSR.53.608.614
[23] Nugraha, B. A. and Rahadi, R. A. (2021). Analysis of young generations toward stock investment intention: A preliminary study in an emerging market. Journal of Accounting and Investment, 22(1), 80-103. doi: 10.18196/jai.v22i1.9606
[24] Perica, I. (2021). The mediating role of managerial accounting in non-profit organizations: a structural equation modelling approach. Croatian operational research review, 12(2), 139-149. doi: 10.17535/crorr.2021.0012
[25] Ringle, C. M., Wende, S. and Becker, J.-M. (2024). SmartPLS 4. Bönningstedt: SmartPLS. Retrieved from: https://www.smartpls.com
[26] Sarstedt, M., Hair Jr, J. F. and Ringle, C. M. (2023). "PLS-SEM: indeed a silver bullet"–retrospective observations and recent advances. Journal of Marketing Theory and Practice, 31(3), 261-275. doi: 10.1080/10696679.2022.2056488
[27] Schreiber, J. B., Nora, A., Stage, F. K. , Barlow, E. A. and King, J. (2006). Reporting Structural

Equation Modeling and Confirmatory Factor Analysis Results: A Review. The Journal of Educational Research, 99(6), 323-337. doi: 10.3200/JOER.99.6.323-338

[28] Sharma, P. N., Liengaard, B. D., Hair, J. F., Sarstedt, M. and Ringle, C. M. (2023). Predictive model assessment and selection in composite-based modeling using PLS-SEM: extensions and guidelines for using CVPAT. European Journal of Marketing, 57(6), 1662-1677. doi: 10.1108/EJM-08-2020-0636

[29] Shmueli, G., Sarstedt, M., Hair, J. F., Cheah, J. H., Ting, H., Vaithilingam, S. and Ringle, C. M. (2019). Predictive model assessment in PLS-SEM: guidelines for using PLSpredict. European journal of marketing, 53(11), 2322-2347. doi: 10.1108/EJM-02-2019-0189

[30] Soper, D. S. (2024). A-priori Sample Size Calculator for Structural Equation Models [Software]. Retrieved from: danielsoper.com/statcalc

[31] Vuković, M. (2024). Generational differences in behavioral factors affecting real estate purchase intention. Property Management, 42(1), 86-104. doi: 10.1108/PM-11-2022-0088

[32] Vuković, M. and Pivac, S. (2021). Does financial behavior mediate the relationship between self-control and financial security? Croatian Operational Research Review, 12(1), 27-36. doi: 10.17535/crorr.2021.0003

[33] Yang, M., Mamun, A. A., Mohiuddin, M., Al-Shami, S. S. A. and Zainol, N. R. (2021). Predicting stock market investment intention and behavior among Malaysian working adults using partial least squares structural equation modeling. Mathematics, 9(8), 873. doi: 10.3390/math9080873

[34] Yıldız, O. and Kelleci, A. (2023). (Consistent) PLS-SEM vs. CB-SEM in Mobile Shopping. Istanbul Gelisim University Journal of Social Sciences, 10(2), 649-667. doi: 10.17336/igusbd.1014138

A cost analysis of single-server discouraged arrivals with differentiated vacation queueing model

Srinivasan Keerthana[1,*], **Jaisingh Ebenesar Anna Bagyam**[1] and **Ramachandran Remya**[1]

[1] *Department of Mathematics, Karpagam Academy of Higher Education, Salem - Kochi Highway, Eachanari, Coimbatore, Tamil Nadu, Pincode - 641021, India.*
E-mail: ⟨{keerthanasrini16, ebenesar.j, remyamath22}@gmail.com⟩

Abstract. This paper investigates an M/M/1 queueing system with differentiated vacations and discouraged arrivals, focusing on two types of vacations. The server switches to type I vacation with rate γ_1 when the system is empty during an active state. If no customers are waiting when it returns from a type I vacation, it then switches to a type II vacation with a rate of γ_2. Both vacation times and service duration follow exponential distributions. The study utilises the Probability Generating Function (PGF) technique to derive steady-state solutions for both vacation policies. Furthermore, the research explores relevant performance metrics and provides numerical examples to illustrate the system's behaviour under various conditions. The cost analysis of the M/M/1 differentiated vacation system with discouraged arrival queueing and various aspects of the system's behaviour under different arrival rates (λ, λ_1, λ_2) are discussed.

Keywords: cost analysis, differentiated vacations, discouraged arrivals, PSO algorithm, steady state solution.

Received: February 16,2024; accepted: June 13, 2024; available online: October 7, 2024

DOI: 10.17535/crorr.2024.0012

1. Introduction

Queueing theory is a branch of applied probability theory with numerous applications, including communication networks, manufacturing facilities, and computer systems. In the traditional queueing strategy, the server is always accessible; however, real-world scenarios may arise where the server becomes unreachable.

When the server finishes serving a unit and finds that the system is empty, it is known as a vacation. Queueing systems with server vacations have garnered significant interest from researchers. In a survey, the primary goal, as described by Doshi [10], is to provide a comprehensive understanding of vacations. The paper demonstrates how analysing alternative vacation models becomes more manageable by comprehending the behaviour of these queueing models. Additionally, the available results are applied to a few selected real-life applications. In another study Levy and Yechiali [21], the utilisation of idle time in an M/G/1 queueing system is analyzed. To prevent the server from being completely inactive, additional work is performed during the vacation. Upon completion of the vacation, the server rejoins the main network. The survey Haviv [15] discusses the strategic timing of arrivals. The book Tian and Zhang [23], have explored various vacation model categories due to their wide range of applications in interaction, technological networks, and manufacturing facilities.

*Corresponding author.

Servers can take two types of vacations. If type I vacation is complete and no more customers are present, the server switches to type II vacation; this is defined as a differentiated vacation. The study considers differentiated vacations, vacation interruptions, and impatient customers (balking and reneging). Recursive methods are used to obtain the explicit expression of the probability, followed by sensitivity analysis which are analysed in Bouchentouf and Guendouzi [5]. An infinite single-server Markovian queueing model with both single and multiple vacation policies, as well as working breakdowns, repairs, balking, and reneging, is analysed by Chettouf et al [9] for a customer care centre. A finite capacity multi-server Markovian queueing model with Bernoulli feedback, synchronous multiple vacation policies, and impatient customers is discussed by Bouchentouf et al [4]. Numerous real-world systems, such as contact centres, manufacturing processes, and contemporary information and communication technology networks, have implemented the proposed queueing model. A single-server Markovian feedback queue with variations of different vacation policies, balking, the server's state-dependent reneging, and retention of reneged customers was examined by Bouchentouf et al [3]. In an analysis of a multi-server queue with impatient customers and Bernoulli feedback, Bouchentouf et al [7] considered a variation with multiple vacations. The probability-generating function approach is used to solve differential equations and derive steady-state probabilities using the Chapman-Kolmogorov equations. In Afroun et al [2], an M/M/1/N queueing system with various vacations, Bernoulli feedback, balking, reneging, and retention of the impatient customers, as well as the potential for a server failure and repair, are examined. Using the Q-matrix (infinitesimal generator matrix) approach, the system's steady-state probabilities are determined. A feedback queueing system featuring a form of multiple vacation policy, balking, the server's state-dependent reneging, and the retention of reneged customers were discussed by Cherfaoui et al [8]. Working vacations and vacation interruptions are covered by threshold policies. The system's performance metrics and steady-state probability are derived from the application of the Successive over-Relaxation (SoR) technique. Additionally, a quasi-Newton optimisation technique is used for an optimum analysis. Ultimately, a conclusion, several numerical examples, and a discussion of the future's potential are provided by Kumar et al [19]. Bouchentouf et al [6] establishes a cost optimisation analysis for an M/M/1/N queueing system with differentiated working vacations, Bernoulli schedule vacation interruption, balking and reneging. Suranga sampath and Liu [22] used various analytical tools, including the Laplace transform, probability-generating functions, and explicit mean and variance systems. Transient state probabilities are calculated, and the study explores the system's behaviour.

Vijayashree and Janani [27] evaluates the transient solution of an M/M/1 queue with differentiated vacation. Vijayashree and Ambika [26] analyses the concept of an M/M/1 queue with differentiated vacation, vacation interruption, and customer impatience. The study examines the mean and variance, presenting a comprehensive analysis.

In a separate study Ebenesar and Chandrika [12], a single-server retrial queueing model with Markovian arrival processes for customer arrivals is discussed. The M/G/1 retrial queueing system has two simultaneous vacation modes. Performance metrics and numerical results are discussed. An analysis is carried out by Ebenesar et al [11] to examine the steady-state behaviour of a single-server retrial queueing model that includes server breakdown and frequent vacation. Performance metrics are considered using supplemental variable approaches.

Admission management is discussed in Ebenesar and Chandrika [13] to balance effective system utilisation by providing acceptable performance metrics. The server's condition determines whether each customer may access the system. Accepted customers receive the first necessary service, with the option for a second service or to exit after the service is rendered. During certain periods, arrivals are restricted due to an extended queue, known as discouraged arrivals. The concept is particularly relevant during the pandemic (COVID), when restrictions are imposed on arrivals from other countries and crowded places. Kumar and Sharma [20] explains a finite Markovian single-server queuing model with discouraged arrivals, reneging, and

retention of reneged customers. The steady-state solution is derived. Performance measures are obtained, and special cases of the model are explored.

Hur and Paik [16] explores an M/G/1 queue subject to a regulatory policy with a general server setup time. The arrival rate fluctuates based on the idle, setup, and busy states of the server. The steady-state queue length distribution function and the Laplace-Stieltjes transform of waiting time are derived. In Hassin et al [14], the RASTA phenomenon is explored, where customers decide whether to join or block queues based on the implications of their entry-level choices. In Tian et al [24], a repairable M/M/1 retrial queueing model with setup delays is reviewed. The server is closed down to reduce operating costs after the system is empty, and the system won't start up until a new customers arrives. Steady-state probability, performance measures, and the effect of some parameter costs are evaluated.

Rasheed and Manoharan [1] examines the concept of discouraged arrival in Markovian queuing systems, where the rapid rate of service is controlled based on the number of customers within the system. The study determines the steady-state probability and other performance metrics for this adaptive queueing system.

The differentiated vacation concepts are discussed by various authors. Ibe and Isijola [17] initially proposed differentiated vacations, which are categorised into longer and shorter durations in this paper. The paper then provides numerical examples with different arrival rates. In another study Isijola and Ibe [18], differentiated vacation with vacation interruption is described. According to Vadivukarasi and Kalidass [25], differentiated vacations lead to bulky entry queues. The matrix geometry approach is used to determine stability criteria, and the probability-generating function is employed to determine system size. The PSO approach is used to examine optimal service rates. Our investigation in this article focuses on the new concept of discouraged arrival rates. This study examines the total cost of the suggested model. The PSO method was also used to determine the system's cost-effectiveness.The findings of this article are displayed in the table together with the different arrival rates and discouraged arrival rates of the cost values and expected number of customers.

This paragraph discusses two kinds of vacations, type I and type II, where the type I vacation rate is lower than the type II vacation rate. Section 2 provides a detailed explanation of this model. In Section 3, the transition diagram, the local balance equation, and the probability for the busy state are obtained. Performance measures are described in Section 4, with numerical examples provided in Section 5. A cost analysis is presented in Section 6, and Section 7 introduces the PSO algorithm for the model. Special cases are provided in Section 8. Practical application is explained in Section 9. Conclusion of this model are described in section 10.

2. The System Descriptions

The proposed model operates under the following assumptions:

Arrival Process:

- The queue has an infinite capacity to accommodate customers.
- The customer arrival process follows a Poisson distribution with a rate of λ.

Service Operation:

- Customers are admitted into the system based on the First-Come-First-Serve (FCFS) principle.
- Service times are modelled to follow an exponential distribution with the parameter μ.

Vacation Types:

- In this model, vacations are incorporated into two categories: type I vacation, denoted by γ_1, and type II vacation, denoted by γ_2. As indicated by the relationship $\gamma_2 \leq \gamma_1$, type II vacation occurs less than type I vacation.
- Customers are served by the server when it is in an active state. Upon completing service during an active state, the server immediately transitions to type I vacation.
- After finishing the type I vacation, the server returns to an active state. If a new customer is present, the server provides immediate service; otherwise, the server switches to a type II vacation.

Arrival Rate Restriction during Vacations:

- During vacation periods, customer arrivals are restricted based on the following rates: $\lambda_2 \leq \lambda_1 \leq \lambda$.
- Here, λ_1 represents the discouraged arrival rate during type I vacation, and λ_2 represents the discouraged arrival rate during type II vacation.

3. Steady-State Solution

Let $N(t)$ be the number of customers in the system at time t and $J(t)$ be the state of the service provider at time t as

$$J(t) = \begin{cases} 0, \text{ while the server are in active state,} \\ 1, \text{ if the server is on first vacation,} \\ 2, \text{ if server is on second vacation.} \end{cases} \tag{1}$$

Then $\{(J(t), N(t)), t \geq 0\}$ is a state-space Markov process.

Let $p_{i,j}$ be the probability that the service provider be in the i^{th} state $(i = 0, 1, 2)$ with $j(\geq 0)$ number of customers.

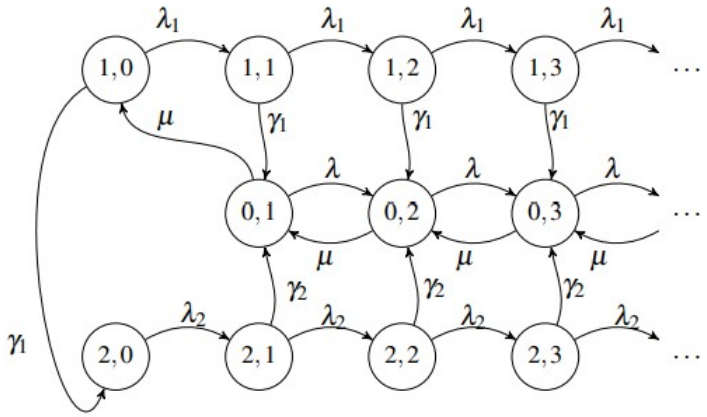


Figure 1: *State transition diagram.*

The steady-state balancing flow equations of the proposed model are as follows:

$$(\lambda+\mu)p_{0,1} = \mu p_{0,2} + \gamma_1 p_{1,1} + \gamma_2 p_{2,1} \tag{2}$$
$$(\lambda+\mu)p_{0,n} = \mu p_{0,n+1} + \gamma_1 p_{1,n} + \gamma_2 p_{2,n} + \lambda p_{0,n-1}, n \geq 2 \tag{3}$$
$$(\lambda_1+\gamma_1)p_{1,0} = \mu p_{0,1} \tag{4}$$
$$(\lambda_1+\gamma_1)p_{1,n} = \lambda_1 p_{1,n-1}, n \geq 1 \tag{5}$$
$$\lambda_2 p_{2,0} = \gamma_1 p_{1,0} \tag{6}$$
$$(\lambda_2+\gamma_2)p_{2,n} = \lambda_2 p_{2,n-1}, n \geq 1 \tag{7}$$

Let,

$$P_i(z) = \sum_{n=1}^{\infty} p_{i,n} z^n, \quad i = 0, 1, 2$$

be the function that generates probabilities of active state and the vacations states.

By summing up all the possible values of n and by multiplying by z^n to equations (1) to (6), we the probability generating functions of active state and vacations states (type I and type II) respectively,

$$P_0(z) = \frac{\mu z p_{0,1} - \gamma_1 z P_1(z) - \gamma_2 z P_2(z)}{\lambda z^2 - (\lambda+\mu)z + \mu} \tag{8}$$
$$P_1(z) = \frac{\lambda_1 z}{\lambda_1(1-z)+\gamma_1} p_{1,0} \tag{9}$$
$$P_2(z) = \frac{\lambda_2 z}{\lambda_2(1-z)+\gamma_2} p_{2,0} \tag{10}$$

Substitute $z = 1$ in equations (7), (8) and (9), we obtain,

$$P_0(1) = \frac{\mu((\gamma_1^2(\lambda_2+\gamma_2)) + (\lambda_1\gamma_2(\lambda_1+\gamma_1)))}{\gamma_1\gamma_2(\lambda_1+\gamma_1)(\mu-\lambda)} p_{0,1} \tag{11}$$
$$P_1(1) = \frac{\lambda_1\mu}{\gamma_1(\lambda_1+\gamma_1)} p_{0,1} \tag{12}$$
$$P_2(1) = \frac{\mu\gamma_1}{\gamma_2(\lambda_1+\gamma_1)} p_{0,1} \tag{13}$$

Finally, by using the rule of total probability,

$$P_0(1) + P_1(1) + p_{1,0} + P_2(1) + p_{2,0} = 1 \tag{14}$$

where

$$p_{1,0} = \frac{\mu}{\lambda_1+\gamma_1} p_{0,1} \tag{15}$$
$$p_{2,0} = \frac{\mu\gamma_1}{\lambda_2(\lambda_1+\gamma_1)} p_{0,1} \tag{16}$$

The probability that the server is in an active state is as follows,

$$p_{0,1} = \frac{\lambda_2\gamma_1\gamma_2(\lambda_1+\gamma_1)(\mu-\lambda)}{\mu\gamma_1^2(\lambda_2+\gamma_2)(\mu+\lambda_2-\lambda) + \mu\lambda_2\gamma_2(\lambda_1+\gamma_1)(\mu+\lambda_1-\lambda)} \tag{17}$$

4. Performance Measures

The expected number of customers when a service provider is in an active state is:

$$E(L_B) = P_0'(1) \tag{18}$$

$$= \frac{\begin{array}{r}\mu^2(((\gamma_1^3)(\lambda_2+\gamma_2)(\lambda_2+\gamma_2))+(\gamma_2^2(\lambda_1+\gamma_1)(\lambda_2+(\lambda_1\gamma_1)))) \\ -\mu(((\lambda\lambda_2\gamma_1^3)(\lambda_2+\gamma_2))+(\lambda_3\gamma_2^2)(\lambda_1+\gamma_1))\end{array}}{\gamma_1^2\gamma_2^2(\lambda_1+\gamma_1)(\mu-\lambda)^2} p_{0,1} \tag{19}$$

The expected number of customers during type I vacation is:

$$E(L_{v_1}) = P_1'(1) \tag{20}$$

$$= \frac{\lambda_1\mu}{\gamma_1^2} p_{0,1} \tag{21}$$

The expected number of customers in the system during type II vacation is:

$$E(L_{v_2}) = P_2'(1) \tag{22}$$

$$= \frac{\mu\gamma_1(\lambda_2+\gamma_2)}{\gamma_2^2(\lambda_1+\gamma_1)} p_{0,1} \tag{23}$$

The total average number of customers in the system is

$$E(L) = E(L_B) + E(L_{v_1}) + E(L_{v_2}) \tag{24}$$

Expected waiting time:

$$E(W) = \frac{E(L)}{\lambda} \tag{25}$$

5. Numerical Analysis

This section presents various numerical examples to illustrate the influence of different parameters, including arrival rate, service rate, and vacation rate, on performance measures.

The relationship between the likelihood of one customer being in an active condition is depicted in Figure 2. With a fixed service rate, $p_{0,1}$ falls whenever λ rises. This figures indicate varying service rates in this instance. Additionally, it emphasised that when service rates rise, $p_{0,1}$ falls. Depending on the service rate, it demonstrates that $p_{0,1}$ increases with λ. That means that whenever the arrival rates increase, the probability that the server will take a type I vacation when the system is empty also increases. Figures 3 and 4 show the impact on $p_{0,1}$ of the discouraging arrival rate during type I and type II vacations, respectively. As the discouraged arrival rate increases during type I vacation, Figure 3 illustrates a decrease in $p_{0,1}$. A decrease in $p_{0,1}$ is also shown in Figure 4 when the rate of discouraged arrivals increases during type II vacation.

The expected number of customers in the system under various situations was evaluated in Figures 5, 6, and 7. As average arrival rates increase, Figure 5 illustrates an increase in the expected number of customers, indicating that arrival rates result in a larger number of customers in the system. The number of customers in the system appears to be decreased by the server's vacation under this policy, as Figure 6 shows the decrease in the expected number of customers during type I vacation. As the discouraged arrival rates during type II vacation increase, Figure 7 shows an increase in the expected number of customers, suggesting that a higher rate of discouraged arrivals during type II vacation results in higher numbers of customers in the system.

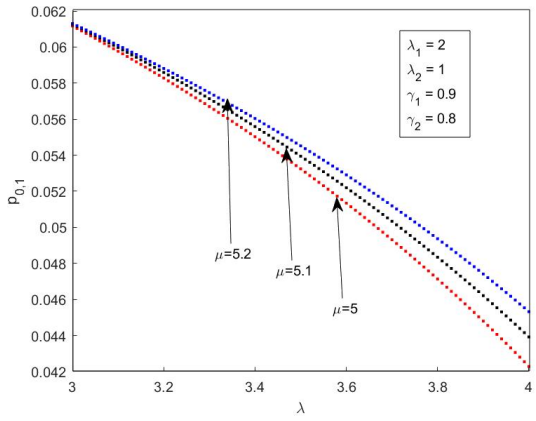


Figure 2: $p_{0,1}$ *against* λ.

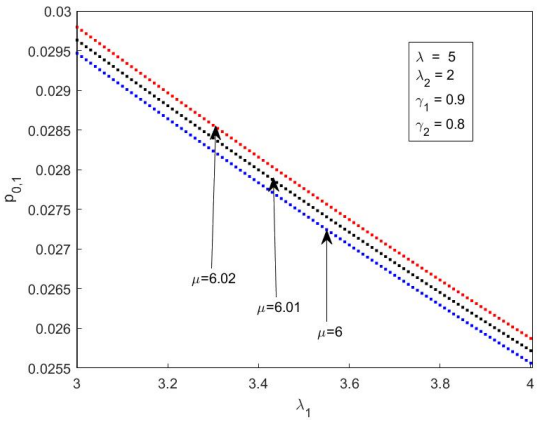


Figure 3: $p_{0,1}$ *against* λ_1.

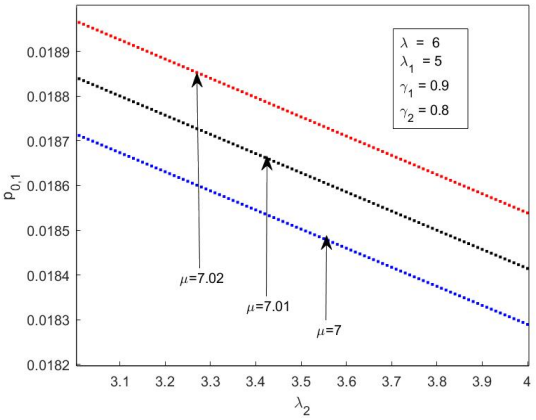


Figure 4: $p_{0,1}$ *against* λ_2.

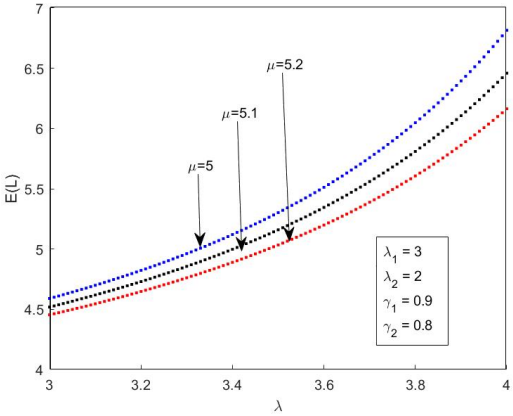


Figure 5: $E(L)$ *against* λ.

The expected waiting time in the system under various situations was evaluated in Figures 8, 9, and 10. As arrival rates increase, Figure 8 illustrates an increase in the expected waiting time with a distinct service rate. As the discouraged arrival rate (during type I vacation) increases, Figure 9 indicates that expected waiting time increases with different service rates. Likewise, as the discouraged arrival rates (during type II vacation) increase, Figure 10 shows expected waiting time increasing with different service rates in the system.

Figure 11 shows that the expected number of customers in the system decreases as the service rate increases with differnet λ. According to this, an increased service rate results in a decrease in customers in the system, which may decrease the waiting time of customers and increase overall effectiveness. Figure 12 shows that the estimated waiting time in the queue decreases for varying arrival rates as the service rate increases. Thus it suggest that longer waiting times for customers are achieved with higher service rates.

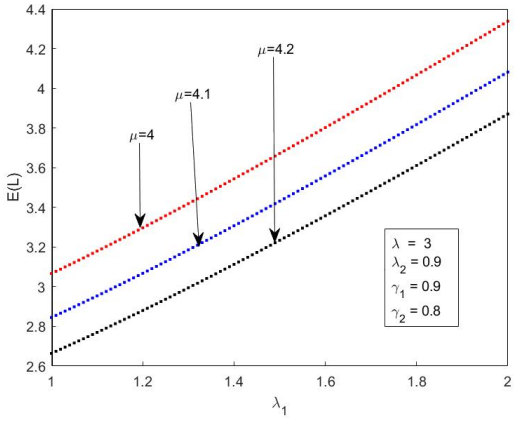


Figure 6: $E(L)$ *against* λ_1.

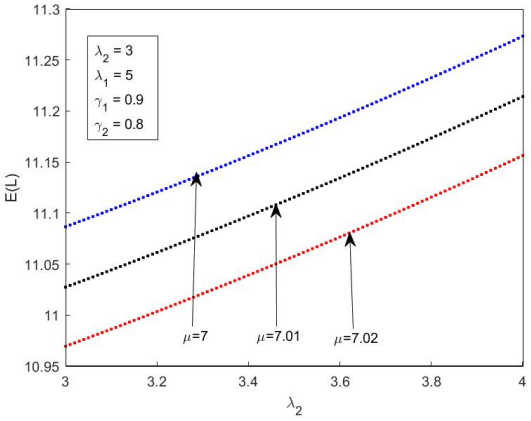


Figure 7: $E(L)$ *against* λ_2.

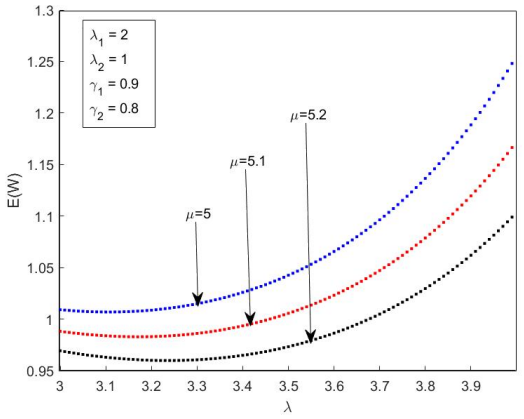


Figure 8: $E(W)$ *against* λ.

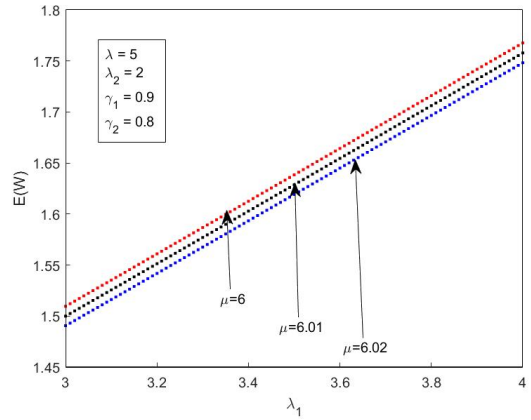


Figure 9: $E(W)$ *against* λ_1.

6. Cost Analysis

Define the cost function TC as

$$TC = C_N E(L) + C_W E(W) + C_0 P_0 + \sum_{i=1}^{2} C_i P_i + C_\mu \tag{26}$$

where,
C_N = holding cost for each customers seen in the system;
C_W = waiting cost if one customers is to receive the service;
C_0 = cost for the period the server handling service process;
C_i = cost when the server is on i^{th} type vacations (i=1,2)
C_μ = cost for service.
$p_{0,n}$ = probability when the server is on active state,
$p_{i,n}$ = probability when the server is on the i^{th} type vacations (i=1,2)
E(L) = the expected number of customers in the system,
E(W) = the expected waiting time of a customers in the system, respectively.

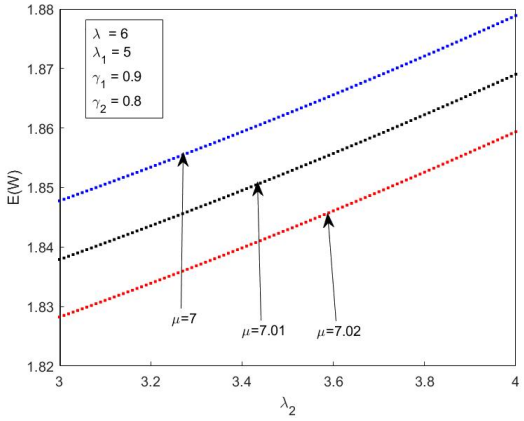


Figure 10: $E(W)$ *against* λ_2.

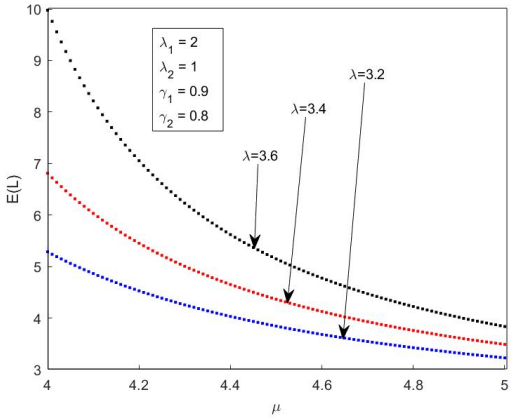


Figure 11: $E(L)$ *against* μ.

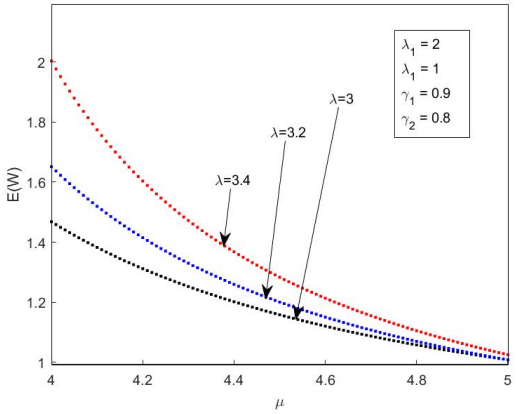


Figure 12: $E(W)$ *against* μ.

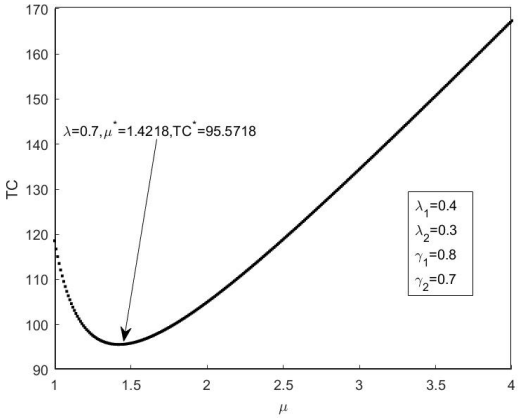


Figure 13: TC *against* λ.

In this section, the cost analysis of the model is studied, as it is very important in designing, improving, and maintaining the model from a cost-benefit point of view. The section 6 Total Cost function (TC) is nonlinear and too complicated to calculate analytically. Numerical optimisation techniques then make it simple to find the ideal value of the service rate. The lowest costs and service rate are shown in Figure 13 using a convex graphical representation. From the figure, it is observed that if the service rate increases, the total cost becomes convex. The expected cost function is optimised by using the PSO method. Figures 14 and 15 show the optimum total cost and service rate with different λ. Since the curves are convex, the presence of ideal values are required. Furthermore, it is noted that, as would be expected intuitively, with λ, both the ideal service in concern and its expected expenses grow.

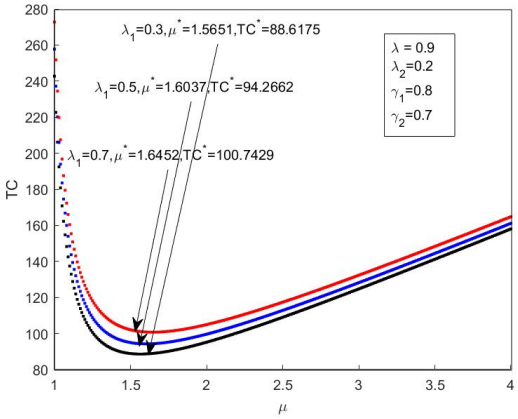


Figure 14: TC *against* λ_1.

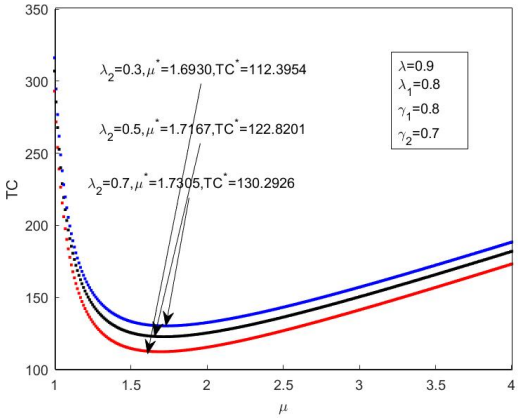


Figure 15: TC *against* λ_2.

7. PSO Algorithm

PSO is a computational method for optimising the service and total cost. One effective technique for figuring out a complicated function's optimal value is the PSO algorithm. It uses location, position and velocity control to find the optimal path based on particle movement. That means minimising the cost corresponding to the best service rate.

When using the PSO algorithm, one can evaluate the service rate behaviour, the expected number of customers, and the cost analysis, as shown in Table 1. The service time, expected number of customers, and cost do increase whenever the arrival rate increases. We consider the arrival rate and discouraged arrival rate on type II vacations to be constant. In type I vacations, the discouraged arrival rate increases, followed by the service rate, the expected number of customers, and the total cost, which are also shown in Table 2. Table 3 shows that in the case that the discouraged arrival rate in type II vacation grows, the corresponding service rate, projected number of customers, and overall cost would all increase. The preceding three tables were analysed using the Particle Swarm Optimisation (PSO) algorithm. The results show that the service rate, expected customer count, and overall cost of the queueing system all increase in tandem with increases in arrival rates and discouraged arrival rates (types I and II). Consider vacation rates and discouraged arrival rates as constants. This observation highlights the sensitivity of these performance metrices to changes in arrival patterns and vacation types, emphasising the importance of accurately modelling and managing these factors in queueing systems to optimise system efficiency and cost-effectiveness. These findings contribute valuable insights to the field of queueing theory. Articulately in understanding the dynamic nature of queueing systems and the implications of varying arrival and vacation rates on system performance.

λ	μ^*	$E(L)$	$E(W)$	TC^*
0.5	1.2272	0.9750	1.9499	94.2419
0.6	1.3224	1.0221	1.7035	94.3628
0.7	1.4218	1.0659	1.5227	95.5718
0.8	1.5238	1.1069	1.3837	97.4617
0.9	1.6275	1.1457	1.2730	99.8042

Table 1: *PSO values for different* λ.

λ_1	μ^*	$E(L)$	$E(W)$	TC^*
0.3	1.56	0.89	0.99	88.61
0.4	1.58	0.95	1.05	91.32
0.5	1.60	1.02	1.13	94.26
0.6	1.62	1.09	1.21	97.41
0.7	1.64	1.17	1.31	100.74

Table 2: *PSO values for different* λ_1.

λ_2	μ^*	$E(L)$	$E(W)$	TC^*
0.3	1.69	1.47	1.63	112.39
0.4	1.70	1.63	1.81	118.20
0.5	1.71	1.76	1.95	122.82
0.6	1.71	1.87	2.08	126.76
0.7	1.73	1.98	2.20	130.29

Table 3: *PSO values for different* λ_2.

8. Special Cases

- While substituting, λ_1, λ_2=λ then the probability coincide with Ibe[2014] as,

$$p_{0,1} = \frac{\lambda\gamma_1\gamma_2(\mu-\lambda)(\lambda+\gamma_1)}{\mu^2(\lambda\gamma_2(\lambda+\gamma_1)+\gamma_1^2(\lambda+\gamma_2))} \tag{27}$$

- Putting λ_2=λ and $\gamma_1 \to \infty$ the the PGF becomes,

$$P_2(z) = \frac{\mu z}{\lambda(1-z)+\gamma_2}p_{0,1} \tag{28}$$

$$P_0(z) = \frac{\mu z(z-1)(\lambda+\gamma_2)}{(\lambda(1-z)+\gamma_2)(-\lambda z^2+(\lambda+\mu)z)}p_{0,1} \tag{29}$$

$$p_{0,1} = \frac{\lambda\gamma_2(\mu-\lambda)}{\mu^2(\lambda+\gamma_2)} \tag{30}$$

The above results coincides with M/M/1 single vacation queueing system.

- Putting $\gamma_2 \to \infty$ and $\lambda_2 \to \infty$ the the PGF becomes,

$$P_0(z) = \frac{\mu z(\lambda_1(z-1)(2\gamma_1+\lambda_1)-\gamma_1^2)}{(1-z)((\lambda z(\lambda+\gamma_1)(\gamma_1+\lambda_1(1-z)))+(\mu(\lambda_1+\gamma_1)(\lambda(z-1)-\gamma_1)))} p_{0,1} \quad (31)$$

$$P_1(z) = \frac{\lambda_1 \mu z}{(\lambda_1+\gamma_1)(-\lambda_1 z+\lambda_1+\gamma_1)} p_{0,1} \quad (32)$$

$$p_{0,1} = \frac{\gamma_1(\lambda_1+\gamma_1)(\mu-\lambda)}{\mu(\gamma_1^2+(\lambda_1+\gamma_1)(\mu+\lambda_1-\lambda))} \quad (33)$$

The above results coincides with M/M/1 single vacation with discouraged arrival queueing system.

9. Practical Application

In many real-world situations, the service facility has been safeguarded from having to wait a long time. Long wait times might discourage potential customers and force servers to improve the quality of their services. Thus, while bearing in mind that queueing systems are state-dependent, it is beneficial to do studies on them. To prevent lengthy queues from accumulating in computer and communication systems, for instance, the congestion management mechanism adjusts packet transmission rates based on the length of the queue at the source or destination. In this case, the arrival rate will vary according to the status of the server. It is more useful for waiting length reduced and provide more convenient.

10. Conclusion

This paper has examined the M/M/1 differentiated vacation queue with a discouraged arrival rate, utilising the probability-generating function approach to formulate the queueing model. Steady-State Probabilities and Performance Measures are also derived in this paper. The investigation into total cost for arrival rate λ, along with discouraged arrival rates λ_1 and λ_2, has provided valuable insights.

Future research will expand upon these findings by investigating an M/M/C discouraged arrival queueing model with varying arrival rates. This extension will further enhance the understanding of queueing systems with differentiated vacations and discouraged arrivals, offering potential applications in various real-world scenarios.

Acknowledgements

The authors would like to thank the editor and the reviewers for pointing out the suggestions for improving the clarity of the paper.

References

[1] Abdul Rasheed, K. V. and Manoharan, M. (2016). Markovian Queueing System with Discouraged Arrivals and Self-Regulatory Servers. Advances in Operations Research. doi: 10.1155/2016/2456135

[2] Afroun, F., Aïssani, D., Hamadouche, D. and Boualem, M. (2018). Q-matrix method for the analysis and performance evaluation of unreliable M/M/1/N queueing model. Mathematical Methods in the Applied Sciences. doi: 10.1002/mma.5119

[3] Bouchentouf, A. A., Boualem M., Cherfaoui, M. and Medjahri L. (2021). Variant vacation queueing system with Bernoulli feedback, balking and server's states-dependent reneging. Yugoslav Journal of Operations Research, 31(4), 557-575. doi: 10.2298/YJOR200418003B

[4] Bouchentouf, A. A., Cherfaoui M. and Boualem, M. (2020). Analysis and Performance Evaluation of Markovian Feedback Multi-Server Queueing Model with Vacation and Impatience. American Journal of Mathematical and Management Sciences, 40(3), 261-282.doi: 10.1080/01966324.2020.1842271

[5] Bouchentouf, A. A. and Guendouzi, A. (2018). Sensitivity analysis of feedback multiple vacation queueing system with differentiated vacations, vacation interruptions and impatient customers. International Journal of Applied Mathematics and Statistics, 57(6), 104-121.

[6] Bouchentouf, A. A., Guendouzi A. and Majid, S. (2020). On impatience in Markovian M/M/1/N/DWV queue with vacation interruption. Croatian Operational Research Review, 11, 21-37. doi: 10.17535/crorr.2020.0003

[7] Bouchentouf A. A., Medjahri L., Boualem M. and Kumar A. (2022). Mathematical analysis of a Markovian multi-server feedback queue with a variant of multiple vacations, balking and reneging. Discrete and continuous models and applied computational science, 30(1), 21-38. doi: doi: 10.22363/2658-4670-2022-30-1-21-38

[8] Cherfaoui, M., Bouchentouf, A. A. and Boualem, M. (2023). Modelling and simulation of Bernoulli feedback queue with general customers' impatience under variant vacation policy. International Journal of Operational Research, 46(4). doi: 10.1504/IJOR.2023.129959

[9] Chettouf, A. A., Bouchentouf, A. A. and Boualem, M. (2024). A Markovian Queueing Model for Telecommunications Support Center with Breakdowns and Vacation Periods. Operations Research Forum, 5(22). doi: 10.1007/s43069-024-00295-y

[10] Doshi, B. T. (1986). Queueing systems with vacations - a survey. Queueing Systems: Theory and Applications, 1(1), 29-66. doi: 10.1007/BF01149327

[11] Ebenesar, A. B., Suganthi, P. and Visali, P. (2014). Unreliable single server retrial queueing model with repeated vacation. International Journal of Mathematics in Operational Research, 26(3), 357-372.doi: 10.1504/IJMOR.2023.134838

[12] Ebenesar, A. B. and Chandrika, U.K. (2010). Single Server Retrial Queuing System with Two Different Vacation Policies. International Journal of Contemporary Mathematical Sciences, 5(32), 1591 - 1598.

[13] Ebenesar, A. B. and Chandrika, U.K. (2018). Fuzzy analysis of bulk arrival two phase retrial queue with vacation and admission control. The Journal of Analysis, 27, 209-232. doi: 10.1007/s41478-018-0118-1

[14] Hassin, R., Haviv, M. and Oz, B. (2023). Strategic behavior in queues with arrival rate uncertainty. European Journal of Operational Research, 309(1), 217-224. doi: 10.1016/j.ejor.2023.01.015

[15] Haviv, M. (2021). A survey of queueing systems with strategic timing of arrivals. 163-198.doi: 10.48550/arXiv.2006.12053

[16] Hur, S. and Paik, S.-J. (1999). The effect of different arrival rates on the N-policy of M/G/1 with server setup. Quality Technology and Quantitative Management, 23(4), 289-299. doi: 10.1016/S0307-904X(98)10088-4

[17] Ibe, O. C. and Isijola, O. A. (2014). M/M/1 Multiple Vacation Queueing Systems with Differentiated Vacations. Modelling and Simulation in Engineering. doi: 10.1155/2014/158247

[18] Isijola-Adakeja, O. A. and Ibe, O. C. (2014). M/M/1 Multiple Vacation Queueing Systems With Differentiated Vacations and Vacation Interruptions. Modelling and Simulation in Engineering. doi: 10.1109/ACCESS.2014.2372671

[19] Kumar, A. Boualem, M., Bouchentouf A. A. and Savita (2022). Optimal Analysis of Machine Interference Problem with Standby, Random Switching Failure, Vacation Interruption and Synchronized Reneging. Applications of Advanced Optimization Techniques in Industrial Engineering. doi: 10.1201/9781003089636-10

[20] Kumar, R. and Sharma, S. K. (2013). A Single Server Markovian Queueing system With Discouraged Arrivals and Retention of Reneged Customers. Yugoslav Journal of Operations Research, 23(2).doi: 10.2298/YJOR120911019K

[21] Levy, Y. and Yechiali, U. (1975). Utilization of idle time in an M/G/1 queueing system. Management Science, 22(2), 202-211. doi: 10.2307/2629709

[22] Suranga Sampath, M. I. G. and Liu, J. (2020). Impact of customers impatience on an $M/M/1$ queueing system subject to differentiated vacations with a waiting server. Quality Technology and Quantitative Management, 17(2), 125-148. doi: 10.1080/16843703.2018.1555877

[23] Tian, N. and Zhang, Z. G. (2006). Vacation Queueing Models: Theory and Applications. Springer. doi: 10.1007/978-0-387-33723-4

[24] Tian, R., Wu, X., He, L. and Han, Y. (2023). Strategic Analysis of Retrial Queues with Setup Times, Breakdown and Repairs. Discrete Dynamics in Nature and Society. doi: 10.1155/2023/4930414

[25] Vadivukarasi, M. and Kalidass, K. (2022). A discussion on the optimality of bulk entry queue with differentiated hiatuses. Operations Research and Decisions, 32(2), 137-150. doi: 10.37190/ord220209

[26] Vijayashree, K. V. and Ambika, K. (2021). An $M/M/1$ Queue subject to Differentiated Vacation with partial interruption and customer impatience. Quality Technology & Quantitative Management, 18(6), 657-682. doi: 10.1080/16843703.2021.1892907

[27] Vijayashree, K. V. and Janani, B. (2017). Transient analysis of an M/M/1 queueing system subject to differentiated vacations. Quality Technology & Quantitative Management, 15(6), 730-748. doi: 10.1080/16843703.2017.1335492

ANFIS computing and cost optimization of an $M/M/c/M$ queue with feedback and balking customers under a hybrid hiatus policy

Aimen Dehimi [1], **Mohamed Boualem**[2,*], **Sami Kahla**[3] **and Louiza Berdjoudj**[4]

[1] *University of Bejaia, Faculty of Exact Sciences, Applied Mathematics Laboratory, 06000 Bejaia, Algeria*
E-mail: ⟨aimen.dehimi@univ-bejaia.dz⟩

[2] *University of Bejaia, Faculty of Technology, Research Unit LaMOS, 06000 Bejaia, Algeria*
E-mail: ⟨mohammed.boualem@univ-bejaia.dz⟩

[3] *Research Center in Industrial Technologies, P.O. Box 64, 16014 Cheraga, Algeria*
E-mail: ⟨samikahla40@yahoo.com⟩

[4] *University of Bejaia, Faculty of Exact Sciences, Research Unit LaMOS, 06000 Bejaia, Algeria*
E-mail: ⟨louiza.berdjoudj@univ-bejaia.dz⟩

Abstract. The present investigation studies a hybrid hiatus policy for a finite-space Markovian queue, incorporating realistic features such as Bernoulli feedback, multiple servers, and balking customers. A hybrid hiatus policy combines both a working hiatus and a complete hiatus. As soon as the system becomes empty, the servers switch to a working hiatus. During a working hiatus, the servers operate at a reduced service rate. Upon completion of the working hiatus and in the absence of waiting customers, the servers enter a complete hiatus. Once the complete hiatus period concludes, the servers resume normal operations and begin serving waiting customers. In the context of Bernoulli feedback, the dissatisfied customer can re-enter the system to receive another service. By utilizing the Markov recursive approach, we examined the steady-state probabilities of the system and queue sizes and other queueing indices, viz. Average queue length, average waiting time, throughput, etc. Using the Quasi-Newton method, a cost function is developed to determine the optimal values of the system's decision variables. Furthermore, a soft computing approach based on an adaptive neuro-fuzzy inference system (ANFIS) is employed to validate the accuracy of the obtained results.

Keywords: ANFIS computing, feedback multi-server queue, hybrid hiatus policy, optimization, recursive approach

Received: July 9, 2024; accepted: August 14, 2024; available online: October 7, 2024

DOI: 10.17535/crorr.2024.0013

1. Introduction

Queueing models with server vacations find extensive applications across various real-life systems such as telecommunications, data and voice transmission networks, and production systems. Over the past several decades, significant research efforts have been dedicated to these models, resulting in comprehensive surveys and seminal works [4, 10, 16, 22] and references therein.

One notable advancement in this area is the introduction of working vacation policies, where servers continue to operate at reduced rates during vacation periods. This concept was first proposed by [20], marking a pivotal development in queueing theory. Extensive literature has

*Corresponding author.

since explored various queueing models incorporating working vacations across diverse contexts [1, 2, 8, 11]. Another significant scenario is queueing models with vacations under balking, prevalent in manufacturing systems, call centers, and transportation networks. Recent research has focused on multi-server systems with impatient customers under both multiple and single vacation policies [16], bulk arrival queueing models with variant working vacation and impatience [6], a finite-capacity discrete-time multi-server queue with synchronous single and multiple working vacations, Bernoulli feedback, and impatient customers [25], and differentiated working vacation policies with impatient customers in single-server queues [7].

In the realm of multi-server queueing models with vacation, two primary types exist: synchronous vacations where all servers take vacations simultaneously [3, 5, 18], and asynchronous vacations where servers take vacations independently [14, 17]. However, despite their practical relevance, these systems are complex, and there remains a gap in their detailed analysis. In this study, we introduce a novel operational policy known as the hybrid hiatus, as proposed by Vadivukarasi and Kalidass [23]. This policy involves servers alternating between two operational states: a working hiatus period, where they operate at reduced capacity, and a complete hiatus period, where no services are provided. The decision to switch between these states depends on real-time queue dynamics and system conditions. For instance, in a hospital emergency department, during a working hiatus, the department operates with reduced staff. If patient demand is low, the department may temporarily close until demand increases, or continue operating at a reduced capacity if immediate care is needed. This policy aims to enhance resource efficiency and responsiveness in dynamic service environments.

To tackle the complexities of these systems, intelligent systems like the Adaptive Neuro-Fuzzy Inference System (ANFIS) have been employed. ANFIS integrates fuzzy logic with neural networks to model nonlinear systems effectively. It has been widely applied for categorization, prediction, control, and optimization tasks, including transient analysis in queueing systems. Initially proposed by [15], ANFIS has made significant contributions to queueing theory [9, 12, 13, 21, 24], enabling researchers to compare analytical formulas with ANFIS-generated numerical outcomes for enhanced system understanding.

In this investigation, we study a finite-capacity Markovian multi-server queue with balking and feedback, governed by a hybrid hiatus policy consisting of a working hiatus and a complete hiatus. When the system becomes empty, the servers transition to a working hiatus, where they serve customers at a reduced rate. Upon completing the working hiatus and with no waiting customers, the servers opt for a complete hiatus. Once the complete hiatus concludes, the servers return to their normal operational state to serve waiting customers.Using the Markov recursive approach, we analyze the steady-state probabilities of the system and queue sizes, along with various queueing metrics such as the expected number of customers in the system and queue, expected waiting times, expected balking rates, and probabilities associated with different server states. We develop a cost function to optimize the system's decision variables using the Quasi-Newton method. Furthermore, we employ a soft computing approach based on an adaptive neuro-fuzzy inference system (ANFIS) to validate the accuracy of our findings.

The structure of this paper is outlined as follows: Section 2 provides a model description and associated mathematical assumptions. In Section 3, we establish the steady-state solution of the model using the recursive method. Section 4 presents explicit formulas for queueing metrics and discusses the ANFIS approach. In Section 5, we introduce the cost model formulation. Section 6 includes numerical illustrations and discusses cost optimization. Finally, Section 7 presents general conclusions and perspectives.

2. Mathematical formulation of the model

We consider a finite capacity multi-server $M/M/c/M$ queueing system with balking customers, hybrid hiatus, and feedback. Key assumptions underlying this model include:

- Customers enter the system following a Poisson process with rate λ.
- During normal busy periods, service times are exponentially distributed with rate β.
- Service times slow down during working hiatus periods, modeled by an exponential distribution with rate α ($\alpha < \beta$).
- Customers are served based on the FCFS (First-Come-First-Served) discipline, and the system has a finite capacity, denoted as M, with c servers.
- Upon arrival, a customer finds the system in one of several states: on hiatus (no servers available), during a normal busy period, or during a working hiatus. The customer decides to either join the queue with probability κ or balk with probability $\kappa' = 1 - \kappa$.
- A hybrid hiatus involves both a working hiatus (WH) and a complete hiatus (CH). When the system becomes empty, servers transition to WH where they operate at a reduced service rate. After completing the working hiatus and if there are waiting customers, servers return to normal busy mode to serve them. If no customers are waiting, servers move to a complete hiatus. Once the CH concludes, servers return to normal operation to attend to any waiting customers.
- If a customer is dissatisfied with the service provided, they have two options: they can leave the system with a probability q, or return later with a probability $q' = 1 - q$. Feedback customers returning later are treated as new arrivals in the system.

The introduced variables are independent of each other.

2.1. Practical motivations

Several key operational dynamics are explored in this study of an $M/M/c/M$ queueing system applied to a hospital emergency department scenario. Patients arrive according to a Poisson process, seeking medical attention serviced with exponentially distributed times under normal conditions (β), and slower times during working hiatuses ($\alpha < \beta$). The department has a finite capacity M, and patients may balk upon arrival if all treatment rooms are occupied, governed by a probability κ. Hybrid hiatuses are implemented, where during working hiatus periods, the department operates at reduced capacity and may transition to complete hiatus if no patients are waiting. Otherwise, it will return to its usual busy state and start serving patients. Patients dissatisfied with wait times or the quality of care can choose to leave (with a probability q) or return later (with a probability $q' = 1 - q$), treated as new arrivals upon their return. This model provides insights into optimizing emergency department operations by managing patient flow, resource utilization, and service quality in dynamic healthcare environments.

3. Steady-state Solution

Let us consider the bivariate process $\{(A(t), N(t)), t \geq 0\}$, where $A(t)$ denotes the number of customers in the system at time t, and $N(t)$ represents the state of the servers at time t, taking one of three values: $N(t) = 0$ when the servers are in normal busy period at time t, $N(t) = 1$ when the servers are in working hiatus period at time t, and $N(t) = 2$ when the servers are in complete hiatus period at time t.

The joint probability $P_{m,j} = \lim_{t\to\infty} P\{A(t) = m, N(t) = j, (m, j) \in \Omega\}$ denotes the steady-state probabilities of the system. Figure 1 depicts the transition diagram of the considered model.

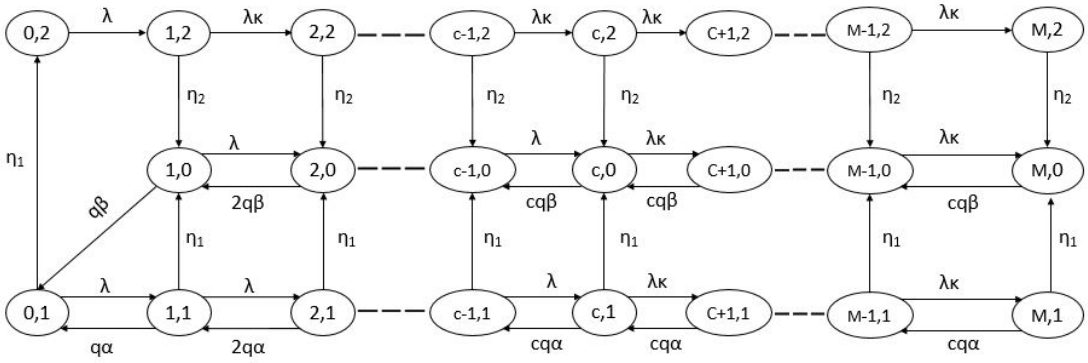


Figure 1: *State transition rate diagram.*

Using the principle of balance equations

$$\lambda P_{0,2} = \eta_1 P_{0,1}, \qquad m = 0, \tag{1}$$

$$(\lambda\kappa + \eta_2)P_{1,2} = \lambda P_{0,2}, \qquad m = 1, \tag{2}$$

$$(\lambda\kappa + \eta_2)P_{m,2} = \lambda\kappa P_{m-1,2}, \qquad 2 \le m \le M-1, \tag{3}$$

$$\lambda\kappa P_{M-1,2} = \eta_2 P_{M,2}, \qquad m = M, \tag{4}$$

$$(\lambda + q\beta)P_{1,0} = 2q\beta P_{2,0} + \eta_1 P_{1,1} + \eta_2 P_{1,2}, \ \ m = 1, \tag{5}$$

$$(\lambda + mq\beta)P_{m,0} = \lambda P_{m-1,0} + (m+1)q\beta P_{m+1,0} + \eta_1 P_{m,1} + \eta_2 P_{m,2}, \ \ 2 \le m \le c-1, \tag{6}$$

$$(\lambda\kappa + cq\beta)P_{c,0} = \lambda P_{c-1,0} + cq\beta P_{c+1,0} + \eta_1 P_{c,1} + \eta_2 P_{c,2}, \tag{7}$$

$$(\lambda\kappa + cq\beta)P_{m,0} = \lambda\kappa P_{m-1,0} + cq\beta P_{m+1,0} + \eta_1 P_{m,1} + \eta_2 P_{m,2}, \ \ c+1 \le m \le M-1, \tag{8}$$

$$cq\beta P_{M,0} = \lambda\kappa P_{M-1,0} + \eta_1 P_{M,1} + \eta_2 P_{M,2}, \tag{9}$$

$$(\lambda + \eta_1)P_{0,1} = \alpha q P_{1,1} + q\beta P_{1,0}, \ \ m = 0, \tag{10}$$

$$(m\alpha q + \lambda + \eta_1)P_{m,1} = \lambda P_{m-1,1} + (m+1)q\alpha P_{m+1,1}, \ \ 1 \le m \le c-1, \tag{11}$$

$$(\lambda\kappa + cq\alpha + \eta_1)P_{c,1} = \lambda P_{c-1,1} + cq\alpha P_{c+1,1}, \tag{12}$$

$$(\lambda\kappa + cq\alpha + \eta_1)P_{m,1} = \lambda\kappa P_{m-1,1} + c\alpha q P_{m+1,1}, \ \ c+1 \le m \le M-1, \tag{13}$$

$$(cq\alpha + \eta_1)P_{M,1} = \lambda\kappa P_{M-1,1}, \tag{14}$$

The normalizing condition is

$$\sum_{m=0}^{M} (P_{m,0} + P_{m,1} + P_{m,2}) = 1. \tag{15}$$

Now, we present the solution of the equations above in the following theorem.

Theorem 1. *The probabilities describing the system size in different operational periods, namely the hiatus period ($P_{m,2}$), working hiatus period ($P_{m,1}$), and normal busy period ($P_{m,0}$), in the steady-state are respectively expressed as follows:*

$$P_{m,2} = \Lambda_m P_{M,2} = \Lambda_m \left(\sum_{m=0}^{M} (\Lambda_m + \theta_1 \chi_m) + \sum_{m=1}^{M} (\theta_2 \Upsilon_m - \delta_m) \right)^{-1}, \ m = 0, 1, 2, ..., M, \tag{16}$$

$$P_{m,1} = \theta_1 \chi_m P_{M,2}, \tag{17}$$

$$P_{m,0} = (\theta_2 \Upsilon_m - \delta_m) P_{M,2}, \tag{18}$$

where

$$\Lambda_m = \begin{cases} 1, & m = M, \\ \frac{\eta_2}{\lambda\kappa}, & m = M-1, \\ \frac{\lambda\kappa+\eta_2}{\lambda\kappa}\Lambda_{m+1}, & 0 \le m \le M-2, \end{cases} \tag{19}$$

$$\chi_m = \begin{cases} 1, & m = M, \\ \frac{cq\alpha+\eta_1}{\lambda\kappa}, & m = M-1, \\ \frac{\lambda\kappa+cq\alpha+\eta_1}{\lambda\kappa}\chi_{m+1} - \frac{cq\alpha}{\lambda\kappa}\chi_{m+2}, & c \le m \le M-1, \\ \frac{\lambda\kappa+(m+1)q\alpha+\eta_1}{\lambda}\chi_{m+1} - \frac{(m+1)q\alpha}{\lambda}\chi_{m+2}, & m = c-1, \\ \frac{\lambda+(m+1)q\alpha+\eta_1}{\lambda}\chi_{m+1} - \frac{(m+2)q\alpha}{\lambda}\chi_{m+2}, & 0 \le m \le c-2, \end{cases} \tag{20}$$

$$\theta_1 = \frac{\lambda\Lambda_0}{\eta_1\chi_0}, \tag{21}$$

$$\Upsilon_m = \begin{cases} 1, & m = M, \\ \frac{cq\beta}{\lambda\kappa}, & m = M-1, \\ \frac{\lambda\kappa+cq\beta}{\lambda\kappa}\Upsilon_{m+1} - \frac{cq\beta}{\lambda\kappa}\Upsilon_{m+2}, & c \le m \le M-1, \\ \frac{\lambda\kappa+(m+1)q\beta}{\lambda}\Upsilon_{m+1} - \frac{(m+1)q\beta}{\lambda}\Upsilon_{m+2}, & m = c-1, \\ \frac{\lambda+(m+1)q\beta}{\lambda}\Upsilon_{m+1} - \frac{(m+2)q\beta}{\lambda}\Upsilon_{m+2}, & 1 \le m \le c-2, \end{cases}$$

$$\delta_m = \begin{cases} 0, & m = M, \\ \frac{\eta_1\theta_1+\eta_2}{\lambda\kappa}, & m = M-1, \\ \frac{\theta_1\eta_1\chi_{m+1}+\eta_2\Lambda_{m+1}}{\lambda\kappa}, & c \le m < M-1, \\ \frac{\theta_1\eta_1\chi_{m+1}+\eta_2\Lambda_{m+1}}{\lambda}, & m = c-1, \\ \frac{\theta_1\eta_1\chi_{m+1}+\eta_2\Lambda_{m+1}}{\lambda}, & 1 \le m \le c-2, \end{cases}$$

$$\theta_2 = \frac{\theta_1(\lambda+\eta_1)\chi_0 - \theta_1 q\alpha\chi_1 + q\beta\delta_1}{q\beta\Upsilon_1}, \tag{22}$$

and

$$P_{M,2} = \left(\sum_{m=0}^{M} (\Lambda_m + \theta_1\chi_m) + \sum_{m=1}^{M} (\theta_2\Upsilon_m - \delta_m) \right)^{-1}. \tag{23}$$

4. Metrics of system performance

▷ The probabilities associated with different server states–normal busy period, working hiatus, and hiatus–are defined as follows:

$$P_{rb} = P_{M,2} \sum_{m=1}^{M} (\theta_2 \Upsilon_m - \delta_m), \tag{24}$$

$$P_{wh} = \theta_1 P_{M,2} \sum_{m=0}^{M} \chi_m, \tag{25}$$

$$P_h = P_{M,2} \sum_{m=0}^{M} \Lambda_m. \tag{26}$$

▷ The expressions for the expected number of customers in the system (L_s) and in the queue (L_q) are defined as follows:

$$L_s = P_{M,2} \left[\theta_2 \sum_{m=1}^{M} m\Upsilon_m - \sum_{m=1}^{M} m\delta_m + \theta_1 \sum_{m=1}^{M} m\chi_m + \sum_{m=1}^{M} m\Lambda_m \right], \tag{27}$$

$$L_q = P_{M,2} \left[\theta_2 \sum_{m=c}^{M} (m-c)\Upsilon_m - \sum_{m=c}^{M} (m-c)\delta_k + \theta_1 \sum_{m=c}^{M} (m-c)\chi_m + \sum_{m=1}^{M} m\Lambda_k \right]. \tag{28}$$

▷ The expected balking rate:

$$B_r = \lambda P_{M,2} \left[\theta_2 \sum_{m=c}^{M} \kappa^{'} \Upsilon_m - \sum_{m=c}^{M} \kappa^{'} \delta_m + \theta_1 \sum_{m=c}^{M} \kappa^{'} \chi_m + \sum_{m=c}^{M} \kappa^{'} \Lambda_m \right]. \tag{29}$$

▷ The expressions for the expected waiting time of customers in the system (W_s) and in the queue (W_q) are given by:

$$W_s = \frac{L_s}{\lambda'}, \quad \text{where} \quad \lambda^{'} = \lambda - B_r, \tag{30}$$

$$W_q = \frac{L_q}{\lambda'}. \tag{31}$$

4.1. Adaptive neuro-fuzzy inference system

The Adaptive Neuro-Fuzzy Inference System (ANFIS), as proposed in [15], combines the principles of fuzzy logic and neural networks to create a powerful tool capable of modeling complex systems. ANFIS operates on a multilayer architecture using Takagi-Sugeno fuzzy inference rules, allowing it to handle multiple inputs and outputs simultaneously with the aid of fuzzy parameters. This approach enables ANFIS to dynamically learn and interpret intricate patterns in both linear and nonlinear relationships. By employing Gaussian functions for membership and utilizing Sugeno-type systems, ANFIS constructs fuzzy if-then rules that are trained using paired input-output data. This training process ensures ANFIS can swiftly adapt and optimize its performance across diverse applications, including telecommunications, atmospheric research, and traffic management.

5. Cost optimization

To construct the cost model, we consider the following cost elements associated with various events:

- $C_{rb}-$ cost per unit time when the servers are in normal busy period,
- C_h- cost per unit time when the servers are in working hiatus period or on hiatus period,
- C_r- cost per unit time when a customer balks,
- C_β (resp. C_α)$-$ cost per service per unit time during normal busy period (resp. during working hiatus period),
- $C_{s-f}-$ cost per service per unit time for a feedback customer,
- C_b- fixed purchase cost of the server per unit.

Our primary objective is to define the total expected cost per unit time for the system in this context:

$$G(\beta,\alpha) = C_{rb}P_{rb} + C_h(P_{wh} + P_h) + C_rB_r + c\beta C_\beta + c\alpha C_\alpha + cq^{'}(\beta+\alpha)C_{s-f} + cC_b.$$

6. Numerical simulation

This section centers on the numerical evaluation of diverse performance metrics within the proposed queueing model, accomplished through parameter variation. It further illustrates how practitioners can effectively utilize and interpret the resultant findings.

6.1. Performance metrics analysis

In this part, we obtain some various performance measures of interest that are computed under different scenarios by using a MATLAB program.

(η_1,η_2)	L_s	P_{rb}	P_{wh}	P_h
(0.5,0.6)	3.1363	0.6283	0.3586	0.0131
(0.6,0.7)	3.0172	0.6695	0.3161	0.0144
(0.7,0.8)	2.9379	0.7022	0.2824	0.0155
(0.8,0.9)	2.8832	0.7286	0.2550	0.0164
(0.9,1.0)	2.8445	0.7505	0.2323	0.0172

Table 1: *Impact of working hiatus and hiatus rates* (η_1,η_2) *when* $\lambda = 6$, $\kappa = 0.4$, $\beta = 2.5$, $\alpha = 1$, $c = 3$, $M = 12$, $q = 0.7$.

κ	W_s	L_q	B_r
0.1	0.7328	0.0488	1.2949
0.3	0.7711	0.2189	1.1412
0.5	0.8288	0.5123	0.9164
0.7	0.8983	0.9291	0.6146
0.9	0.9787	1.4841	0.2304

Table 2: *Impact of non-balking probability* κ *when* $\lambda = 6$, $\eta_1 = 0.5$, $\eta_2 = 0.8$, $\beta = 2.5$, $\alpha = 1$, $c = 3$, $M = 12$, $q = 0.7$.

Figure 2 presents the Gaussian function used to select fuzzy input parameters, like λ, β, α.

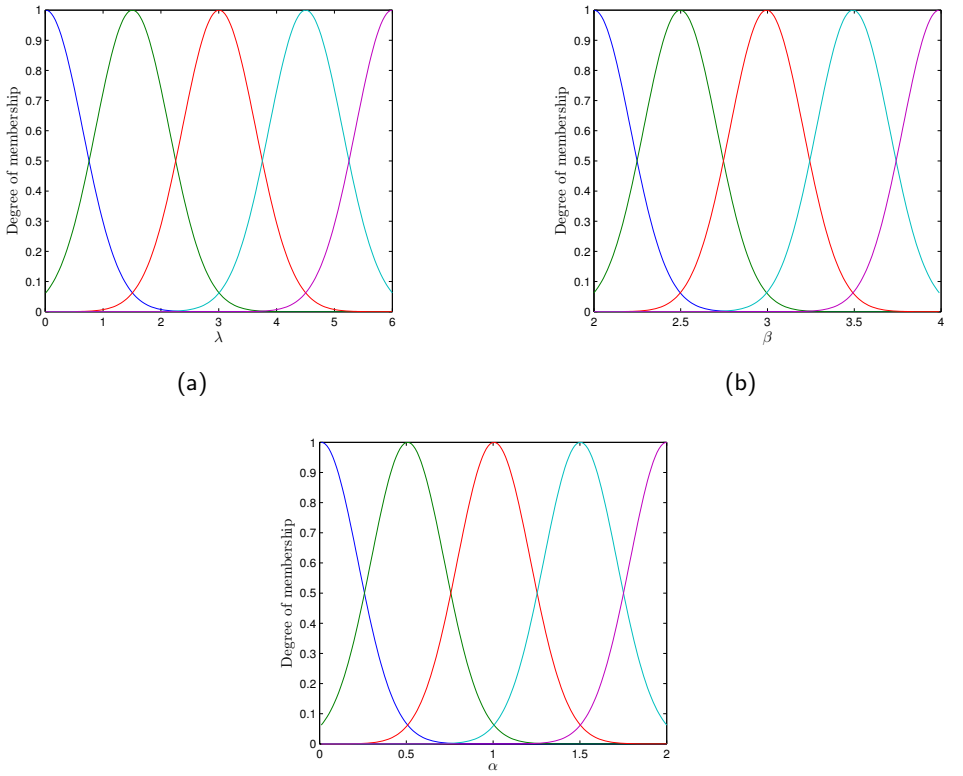


Figure 2: *ANFIS membership function for input variables* λ, β *and* α.

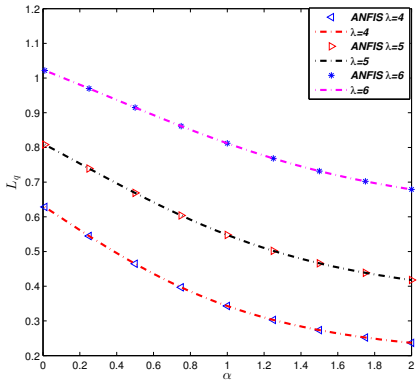


Figure 3: Impact on L_q of α by varying λ.

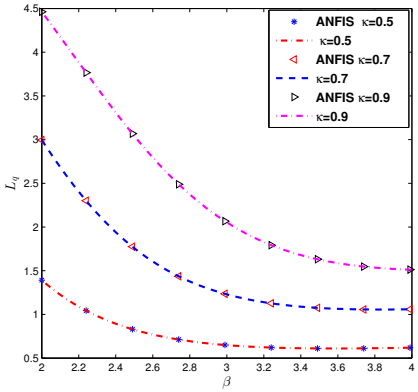


Figure 4: *Impact on* L_q *of* β *by varying* κ.

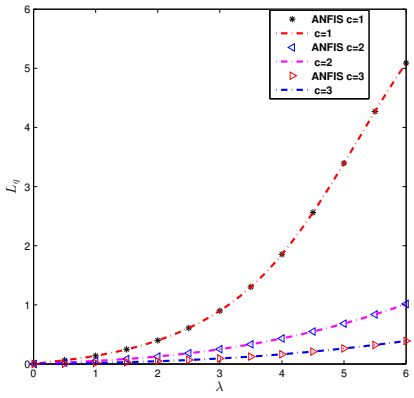


Figure 5: *Impact on L_q of λ by varying c.*

Table 1 illustrates that by increasing the working hiatus rate η_1 and hiatus rate η_2, the system tends to transition more quickly to the normal busy period, increasing the probability of the system being in a normal busy state and decreasing the probability of a working hiatus. Consequently, this results in a decrease in the mean number of customers in the system. Simultaneously, the probability of entering a complete hiatus state increases. This trend is observed as η_1 and η_2 are jointly increased, with η_2 being smaller than η_1.

Table 2 investigates the influence of non-balking probability κ on key performance metrics in a queueing model under fixed parameter settings. As κ increases, the expected waiting time W_s in the system tends to increase, indicating longer waits for customers. Concurrently, the expected number of customers in the queue L_q also rises, reflecting an accumulation of customers waiting for service. Interestingly, the expected balking rate B_r decreases as κ increases, suggesting that higher probabilities of customers entering the system lead to fewer customers choosing to leave without service. These insights underscore the critical role of non-balking probability in shaping customer experience and operational dynamics within the queueing system.

From Figures 3 to 5, we explore the nuanced impacts of key parameters on the average queue length L_q and compare these insights with results derived from the ANFIS model. Figure 3 demonstrates a reduction in L_q as the service rate during working hiatus periods increases, whereas increasing the arrival rates leads to an increase in L_q. Additionally, Figure 4 reveals a decrease in L_q with increasing service rates during normal busy periods, while a decrease in balking probability results in higher L_q. In Figure 5, we observe a rise in L_q with increased arrival rates, whereas increasing the number of servers leads to a decrease in L_q. Comparing these findings with ANFIS model predictions shows a close alignment between both approaches, indicating the reliability and accuracy of the ANFIS model in simulating queue dynamics and validating analytical results.

6.2. Optimal numerical cost

This section aims to determine the optimal service rates β and α that minimize the expected cost function. Due to the complex and non-linear nature of this optimization problem, analytical solutions are impractical. Therefore, we employ advanced nonlinear optimization techniques, specifically the Quasi-Newton method (QNM), to find the optimal values (β^*, α^*) under fixed parameter conditions within the cost model framework.

The optimization problem is formulated as follows:

$$\min_{\beta,\alpha} G(\beta,\alpha)$$
$$\text{s. t. } \begin{cases} \beta > \alpha \\ \alpha > 0. \end{cases}$$

In what follows, the optimal solutions are given by applying the QNM for various system parameters. To do this, we fix the parameters as: $C_{rb} = 80$, $C_h = 60$, $C_r = 50$, $C_\beta = 1$, $C_\alpha = 1$, $C_{s-f} = 1$, $C_b = 1$.

λ	β^*	α^*	$G^*(\beta^*, \alpha^*)$
7	5.2308	4.6689	149.6463
7.5	5.5459	5.0649	155.2061
8	5.8278	5.4562	160.7644
8.5	6.1411	5.8804	166.3211
9	6.4199	6.2789	171.8748

Table 3: *The optimal* (β^*, α^*) *and* $G^*(\beta^*, \alpha^*)$ *for various values of* λ *when* $\kappa = 0.4$, $\eta_1 = 0.6$, $\eta_2 = 1.1$, $c = 4$, $M = 10$, $q = 0.7$.

c	β^*	α^*	$G^*(\beta^*, \alpha^*)$
2	9.1250	8.0762	165.4715
3	6.2465	6.1956	151.7746
4	4.9263	4.2588	144.0853
5	4.0254	3.0062	139.3775
6	3.3214	2.1391	136.3725
7	2.7058	1,3942	133.9343

Table 4: *The optimal* (β^*, α^*) *and* $G^*(\beta^*, \alpha^*)$ *for various values of* c *when* $\kappa = 0.4$, $\eta_1 = 0.6$, $\eta_2 = 1.1$, $\lambda = 6.5$, $M = 10$, $q = 0.7$.

Table 3 illustrates the optimal values (β^*, α^*) and corresponding objective function $G^*(\beta^*, \alpha^*)$ for different arrival rates λ in a specific queueing model. With fixed parameters $\kappa = 0.4$, $\eta_1 = 0.6$, $\eta_2 = 1.1$, $c = 4$, $M = 10$, and $q = 0.7$, the results show that as λ increases from 7 to 9, both β^* and α^* also increase. This indicates that higher arrival rates necessitate larger values of β and α for optimal system performance. Furthermore, $G^*(\beta^*, \alpha^*)$ increases with λ, reflecting improved system performance metrics associated with higher arrival rates. The findings provide critical insights into optimizing queueing systems under varying demand conditions.

From Table 4, increasing the number of servers results in lower minimum costs and service rates, indicating that reducing the server count would be costly. The study optimizes (β, α) and evaluates G across various c values in a finite-capacity Markovian multi-server queue with balking and feedback. Optimal (β^*, α^*) values decrease as c increases, suggesting adjustments to maintain efficiency and customer satisfaction. G^* decreases with higher c, showing enhanced system performance under these conditions, offering strategic insights for managing queueing systems effectively.

7. Conclusion

In this investigation, we explored a finite-capacity Markovian multi-server queue with balking and feedback mechanisms, governed by a hybrid hiatus policy integrating both working and complete hiatus periods. We examined the effectiveness of this policy as servers transitioned from normal operations to a reduced service rate during working hiatus periods, followed by a complete hiatus when no customers were waiting. Once these hiatus concluded, servers resumed normal operations to attend to waiting customers. Using the Markov recursive approach, we analyzed the system's steady-state probabilities and queue metrics, including key measures such as the expected number of customers in the system and queue, expected waiting times, expected balking rate, and probabilities associated with different server states. To optimize decision variables within the system, we developed a cost function implemented through the Quasi-Newton method. Additionally, we validated our results using a soft computing technique, specifically an adaptive neuro-fuzzy inference system (ANFIS), to ensure the robustness and accuracy of our findings. These comprehensive analyses and methodologies provide valuable insights into enhancing operational efficiency and customer satisfaction in complex queuing environments.

The model discussed in the paper can be extended to address more complex scenarios, such as an unreliable multi-server queue with heterogeneous customers, which introduces additional layers of complexity to the problem. Furthermore, the exponential assumptions can be relaxed by incorporating phase-type distributions for service times. These extensions would broaden the applicability of the model, allowing for more realistic simulations and analyses in diverse queueing environments.

Acknowledgements

The authors would like to thank the annoymous reviewers for their valuable comments and suggestions to improve the quality of this paper.

References

[1] Baba, Y. (2005). Analysis of a GI/M/1 queue with multiple working vacations. Operations Research Letters, 33(2), 201-209. doi: 10.1016/j.orl.2004.05.006

[2] Banik, A. D., Gupta, U. C. and Pathak, S. S. (2007). On the $GI/M/1/N$ queue with multiple working vacations. analytic analysis and computation. Applied Mathematical Modelling, 31(9), 1701–1710. doi: 10.1016/j.apm.2006.05.010

[3] Bouchentouf, A. A., Boualem, M., Yahiaoui, L. and Ahmad, H. (2022). A multi-station unreliable machine model with working vacation policy and customers impatience. Quality Technology and Quantitative Management, 19(6), 766–796. doi: 10.1080/16843703.2022.2054088

[4] Bouchentouf, A. A., Cherfaoui, M. and Boualem, M. (2019). Performance and economic analysis of a single server feedback queueing model with vacation and impatient customers. Opsearch, 56(1), 300–323. doi: 10.1007/s12597-019-00357-4

[5] Bouchentouf, A. A., Cherfaoui, M. and Boualem, M. (2021). Analysis and performance evaluation of Markovian feedback multi-server queueing model with vacation and impatience. American Journal of Mathematical and Management Sciences, 40(3), 261–282. doi: 10.1080/01966324.2020.1842271

[6] Bouchentouf, A. A. and Guendouzi, A. (2020). The $M^X/M/c$ Bernoulli feedback queue with variant multiple working vacations and impatient customers: Performance and economic analysis. Arabian Journal of Mathematics, 9(2), 309–327. doi: 10.1007/s40065-019-0260-x

[7] Bouchentouf, A. A., Guendouzi, A. and Majid, S. (2020). On impatience in Markovian $M/M/1/N/DWV$ queue with vacation interruption. Croatian Operational Research Review, 11, 21–37. doi: 10.17535/crorr.2020.0003

[8] Bouchentouf, A. A. and Yahiaoui, L. (2017). On feedback queueing system with reneging and retention of reneged customers, multiple working vacations and Bernoulli schedule vacation interruption. Arabian Journal of Mathematics, 6(1), 1–11. doi: 10.1007/s40065-016-0161-1

[9] Divya, K. and Indhira, K. (2024). Performance analysis and ANFIS computing of an unreliable Markovian feedback queueing model under a hybrid vacation policy. Mathematics and Computers in Simulation, 218, 403-419. doi: 10.1016/j.matcom.2023.12.004

[10] Doshi, B. T. (1986). Single server queues with vacation-A survey. Queueing Systems, 1, 29–66. doi: 10.1007/BF01149327

[11] Ganie, S. M. and Manoharan, P. (2018). Impatient customers in an $M/M/c$ queue with single and multiple synchronous working vacations. Pakistan Journal of Statistics and Operation Research, 571-594.

[12] Indumathi, P. and Karthikeyan, K. (2024). ANFIS-Enhanced $M/M/2$ Queueing Model Investigation in Heterogeneous Server Systems with Catastrophe and Restoration. Contemporary Mathematics, 2482-2502. doi: 10.37256/cm.5220243977

[13] Jain, M. and Meena, R. K. (2018). Vacation model for Markov machine repair problem with two heterogeneous unreliable servers and threshold recovery. Journal of Industrial Engineering International, 14, 143-152. doi: 10.1007/s40092-017-0214-x

[14] Jain, M. and Upadhyaya, S. (2011). Synchronous working vacation policy for finite-buffer multiserver queueing system. Applied Mathematics and Computation, 217(24), 9916–9932. doi: 10.1016/j.amc.2011.04.008

[15] Jang, J. S. (1993) ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans Syst Man Cybern, 23(3), 665–685. doi: 10.1109/21.256541

[16] Kadi, M., Bouchentouf, A. A. and Yahiaoui, L. (2020). On a multiserver queueing system with customers' impatience until the end of service under single and multiple vacation policies. Applications and Applied Mathematics: An International Journal (AAM), 15(2), 4. Retrieved from: digitalcommons.pvamu.edu

[17] Lin, C. H. and Ke, J. C. (2009). Multi-server system with single working vacation. Applied Mathematical Modelling, 33(7), 2967-2977. doi: 10.1016/j.apm.2008.10.006

[18] Prakati, P. (2024). $M/M/C$ queue with multiple working vacations and single working vacation under encouraged arrival with impatient customers. Reliability: Theory and Applications, 19(1 (77), 650-662.

[19] Selvaraju, N. and Goswami, C. (2013). Impatient customers in an $M/M/1$ queue with single and multiple working vacations. Computers and Industrial Engineering, 65(2), 207–215. doi: 10.1016/j.cie.2013.02.016

[20] Servi, L. D. and Finn, S. G. (2002). $M/M/1$ queues with working vacations ($M/M/1/WV$). Performance Evaluation, 50(1), 41–52. doi: 10.1016/S0166-5316(02)00057-3

[21] Sethi, R., Jain, M., Meena, R. K. and Garg, D. (2020). Cost optimization and ANFIS computing of an unreliable $M/M/1$ queueing system with customers' impatience under N-policy. International Journal of Applied and Computational Mathematics, 6, 1-14. doi: 10.1007/s40819-020-0802-0

[22] Tian, N. and Zhang, Z. G. (2006). Vacation queueing models: theory and applications(Vol 93). Springer Science and Business Media.

[23] Vadivukarasi, M. and Kalidass, K. (2022). A discussion on the optimality of bulk entry queue with differentiated hiatuses. Operations Research and Decisions, 32(2), 137-150. doi: 10.37190/ord220209

[24] Upadhyaya, S. and Kushwaha, C. (2020). Performance prediction and ANFIS computing for unreliable retrial queue with delayed repair under modified vacation policy. International Journal of Mathematics in Operational Research, 17(4), 437-466. doi: 10.1504/IJMOR.2020.110843

[25] Yahiaoui, L., Bouchentouf, A. A. and Kadi, M. (2019). Optimum cost analysis for an $Geo/Geo/c/N$ feedback queue under synchronous working vacations and impatient customers. Croatian Operational Research Review, 10, 211-226. doi: 10.17535/crorr.2019.0019

Multi server queuing model with dynamic power shifting and performance efficiency factor

Sreelatha V[1], **Elliriki Mamatha**[1,*], **Chandra S Reddy**[2] **and Krishna Anand S**[3]

[1] *Department of Mathematics, GITAM University, Bengaluru, India*
E-mail: ⟨vsreelat@gitam.in, mellirik@gitam.edu⟩

[2] *Department of Engineering, Garden City University, Bengaluru, India*
E-mail: ⟨saisrimax@gmail.com⟩

[3] *Department of AI&DS, Shridevi Institute of Engineeing and Technoogy, Tumkur, India*
E-mail: ⟨skanand86@gmail.com⟩

Abstract. The need for high performance models in the queuing systems; availability in the fields of computing and communication systems and logistics management poses extensive challenges in design and development of the appropriate modeling. In this work, we propose a robust probabilistic model to mitigate these problems by appropriately choosing a multi-server queuing system with dynamic service facility. The arrival substances, such as packets or jobs or customers or consumers, follow a Poisson arrival process and these arrivals enter a queuing system according to FIFO discipline. The system designed in the system is armed with a fixed number of service stations, in which some servers are capable of allocating additional service capacity by adjusting dynamically. When a customer arrives at the system, if a free server is available, it is immediately served by one of the free processors; if no server is free, that is all are busy, the arrived job is accommodated in the queue and waits for service in the system. In this paper, we proposed a stochastic model to handle peak loads efficiently, boosting service capacity during burst arrivals. Numerical results presented in this paper are generated by the spectral expansion method to demonstrate the model's performance, offering insights into its efficiency and accuracy. Furthermore, we derived some important special cases of speedup factor, which provide the mathematical estimation of system performance in terms of computation and communication times.

Keywords: dynamic service facility and load distribution, mean waiting time, multi-server system, QBD process, spectral expansion method

Received: August 11, 2023; accepted: May 6, 2024; available online: October 7, 2024

DOI: 10.17535/crorr.2024.0014

1. Introduction

In real-life queuing systems, especially during peak periods, it's essential to dynamically adjust server power to accommodate increased customer arrival rates [1, 12, 16]. This flexibility is crucial across various industries, including manufacturing and management. During service time, workload variations are often substantial. At odd times arrivals may be minimal so that servers sit idle during lean periods. Hence, for cost-effectiveness optimizing server utilization is imperative which leads to prolonged lifespan of the servers. Identifying a suitable strategy to allocate power dynamically to the server at the peak period enhances the resources efficiently. It tends to minimize resource wastage, optimize service delivery and responsiveness in queuing systems.

*Corresponding author.

The pursuit of high-performance levels has led to numerous challenges in modeling, designing, and developing fault-tolerant wireless systems [15]. Despite the rapid growth of communication services, customers' expectations for availability and performance remain unchanged [21]. Serving signal/data packets and evaluating their performance involves complex computations, leading to scientific challenges in computing, network communication, and logistics systems [6, 17]. This complexity paves the way for designing models where a single unbounded queue evolves as the environment's state changes over time . Instantaneous service to the customer and its arrival largely depends on the environment's state and, to some extent, on the number of customers arriving at the system. Intrinsically, developing a competent model to accommodate this dynamic service facility is essential to meet consumer requirements and safeguarding system safety and reliability.

Addressing additional power allocation requirements and optimizing the service systems necessitates tackling the recent technology-scaling issues. These service issues are closely connected to the great challenges to handle during peak times [19]. To mitigate these issues Power Shifting mode is one viable solution that works efficiently. In this model, during peak arrival, service capacity is bolstered to selected servers by enhancing its service capacity. This approach, proposed in this paper ensures seamless, uninterrupted time bounded service and optimizes the system performance [5, 18]. To alleviate peak power consumption, one of the strategies available in the field is dynamically increasing service capacity [11]. This can be achieved through integrating activity-related power estimation techniques and real-time performance feedback [20].

This process can be easily extended to a multi-server computer system to solve complex problems. We can observe that dynamically power shifting mode successfully improves power management, alleviating monetary burdens for various industrial organizations.

The contributions of this paper encompass the following:

- Exemplifying the greater efficiency of providing dynamic power allocation over static budgeting.
- Analyzing critical system and workload factors is essential for the success of power shifting proposals.
- Proposing performance-sensitive power budget enforcement mechanisms to ensure system reliability.

Present available optimization models in the multiprocessor queuing system addresses various resource allocation strategies in computing, and communication [13]. Its significance is exemplified through a case study where optimal resource allocation for computing is attained by minimizing energy usage while accounting for constraints such as average response time, response time reliability, queuing system stability, and the maximum allowable quantity of resources [14]. This study underscores the importance of incorporating response time reliability to ensure service quality in cloud computing resource allocation [8].

The model describes a network comprising power-constrained nodes transmitting over channels, such as wireless links with adaptive transmission rates, as outlined in [15, 22]. Customers randomly enter the system and await service in the queue at each node, with their data transmitted through the network to respective destinations. To examine various traffic organization levels, a mathematical model is formulated using rate matrices for support. The design involves power allocation and joint routing distribution to stabilize the system and ensure bounded average service guarantees when input rates fall within the capacity region [23]. This performance holds for both centralized and decentralized implementations, considering general arrival and channel state processes. The network system is monitored, and the system stability of decentralized algorithms is studied concerning a mobile relay strategy.

In the work [2], a dynamic power control problem is addressed, focusing on two similar-level service states subject to random variations in connectivity and switchover server delays between queues. In each time slot, the server determines whether to maintain a constant level of service capacity or switch to an additional power level, thus increasing the service capacity [9]. This decision is based on the current connectivity and queue length information. The introduction of switchover time as a modeling parameter adds a new layer of complexity, enhancing the overall interest in the problem.

To describe system stability, a novel approach is proposed. In this method employs state-action frequencies to identify stationary solutions of the Markov Process and formulate a corresponding structured plan [3, 7]. The stability region is characterized with respect to connectivity parameters. This characterization aids in the development of a new framework for throughput-optimal network policies, supported by state-action frequencies.

2. Mathematical Model and QBD Process System

A queuing network system is modeled in terms of a discrete two-dimensional Markov process on a semi-infinite lattice strip. The process follows a Markovian property, and the transition state of the system at observation time t can be expressed by two random variables $I(t)$ and $J(t)$ used to study the system state at any time t is represented by a two-dimensional random vector. $I(t)$ represent the system's operative state, and its values belong to $[0, N]$ interval of integers, whereas $J(t)$ represents the number of customers present in the queue (including served customers) that may be finite or infinite depending on the queuing system. The Markov process for the QBD queuing system is denoted by: $X(t) = [I(t), J(t); t \geq 0]$ with its state space $[0, 1, 2, ...N] \times [0, 1...]$ for infinite queue size. Let $X(t)$ represents the customer's position at discrete time t with mean arrival and service rates. The number of births at an i^{th} time interval $(t, t + \Delta t)$ with time Δt would be $\sigma\Delta t + o(\Delta t)$.

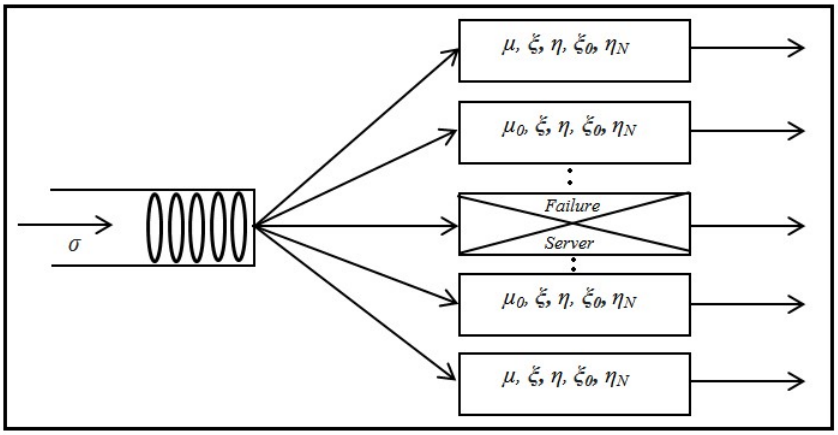


Figure 1: *Parallel processor servicing system with dynamic power.*

Let there be $N + 1$ processor configurations, represented by the values $I(t) = 0, 1, \ldots, N$, denoting operative states of the multiprocessor system. These configurations constitute the operative states of the model, and the model assumptions ensure that $I(t)$, for $t \geq 0$, forms an irreducible Markov process. Subsequently, $X(t) = \{I(t), J(t)]; t \geq 0\}$ represents an irreducible Markov process on a lattice strip (a QBD process) that model the system. This system has been scrutinized for exact performability [4, 10], even under infinite waiting time, i.e., for $L \to \infty$.

Matrix A is the instantaneous transition rates from operative state i to operative state k, with zeros on the main diagonal, indicating purely lateral transitions within the model X. Matrices B and C serve as transition matrices for one-step upward and one-step downward transitions, respectively. Moreover, when the transition rate matrices do not depend on j, the system reaches a steady state for $j \geq M$, where M, an integer constant, represents a threshold value, and the process X evolves through the following instantaneous transitions:

- A_j: Purely lateral transition rate, from state (i, j) to state (k, j), $(i = 0, 1, \ldots, N, k = 0, 1, \ldots, N; i \neq k; j = 0, 1, \ldots, L)$, caused by a change in the operative state (i.e. servers sleeping or break-down followed by service up during arrival, and a repair time).
- B_j: One-step upward transition rate matrix, from state (i, j) to state $(k, j + 1)$, $(i = 0, 1, \ldots, N, k = 0, 1, \ldots, N,$ and $j = 0, 1, \ldots, L)$, caused by a job arrivals into the system.
- C_j: One-step downward transition rate matrix, from state (i, j) to state $(k, j - 1)$, $(i = 0, 1, \ldots, N, k = 0, 1, \ldots, N,$ and $j = 0, 1, \ldots, L)$, caused by the departure of a serviced job.

Let the power be allocated dynamically to the system at the stages where the system needs more power to process customers' delays during peak time. At peak times, it requires more customer service; hence it requires extra power to accomplish the work. During this busy period, additional servers are dynamically assigned to the system. In the regular period, the system operates with a fixed service capacity. The parameter μ is the normal mean service rate and μ_0 is the dynamically allocated power to service for the system during a busy time. It can be observed that $J(t)$ is a process that may move up or down depending on the customer's arrival or departure from the queue. $I(t)$ is the service state corresponding to the arrivals that takes values $0, 1, 2, \ldots, p, q, q+1, q+2, \ldots, N$. Here the states 0 to p represent the system is working in normal mode, whereas from state q to N the system moves work with dynamic mode.
The QBD Markov system can be expressed in mathematical form as:

$$\begin{aligned}
x_1(t+1) &= S_{p+1}x_{p+1}(t) + \ldots + S_{p+q}x_{P+q}(t) - (\mu_1 + \sigma_1)x_1(t) \\
x_2(t+1) &= \sigma_1 x_1(t) - (\mu_2 + \sigma_2)x_1(t) \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
x_{p+q+r-1}(t+1) &= \sigma_{p+q+r-2}(t)x_{p+q+r-2}(t) - (\mu_{p+q+r-1} + \sigma_{p+q+r-1})x_{p+q+r-1}(t) \\
x_{p+q+r}(t+1) &= \sigma_{p+q+r-1}x_{p+q+r-1}(t) - \mu_{p+q+r}x_{p+q+r}(t)
\end{aligned} \tag{1}$$

$x_i(t)$ indicates the system state at i^{th}transient position with arrival and service rates σ_i, μ_i respectively. At the r^{th} state the system moves from the transient state to the steady state, so that the system will not depend on arrivals. Equation (1) can be modeled by the spectral expansion method with transient matrices Aj, Bj, and Cj where the mean arrival rates, the service rates, and the additional dynamic allocations are denoted by σ μ and μ_0, respectively. The system is prone to breakdown either a single server randomly, with a mean rate ξ or bulk servers at a rate ξ_0. The service rates to repair these servers are represented by η and η_N for single server and all servers, respectively. It is explained in Figure 1. The service for the packets arrived at the system is followed by FIFO (first in - first out) discipline. Once the service is completed, packets are dispatched from the system. Matrix A_j is purely phase transitions representing services, and B_j is the upward transitions matrix representing the customers' new arrival. Matrix C_j represents the downward transition matrix. It represents the number of customers serviced during the system up.

As previously stated, matrix A_j represents purely phase transitions representing services, whereas matrix B_j denotes the upward transitions matrix depicting customers' new arrivals. Conversely, matrix C_j represents the downward transition matrix, indicating the number of customers serviced during the system's operation.

$$A_j = \begin{bmatrix} 0 & N\eta_0 & 0 & \cdots & 0 & N\eta_N \\ \xi+\xi_0 & 0 & (N-1)\eta_0 & \cdots & 0 & N\eta_N \\ \xi_0 & N\eta_0 & 0 & \cdots & 0 & N\eta_N \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \xi_0 & N\eta_0 & 0 & \cdots & 0 & \eta_N+\eta_0 \\ \xi_0 & 0 & 0 & \cdots & N\xi & 0 \end{bmatrix} \tag{2}$$

$$B_j = \begin{bmatrix} \sigma & 0 & 0 & \cdots & 0 & 0 \\ 0 & \sigma & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma & 0 \\ 0 & 0 & 0 & \cdots & 0 & \sigma \end{bmatrix} \tag{3}$$

$$C_j = \begin{bmatrix} min(0,j)\mu & 0 & \cdots & \mu_0 & \cdots & \mu_0 & 0 & \cdots & 0 \\ 0 & min(1,j)\mu & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & min(p+1,j)\mu & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & min(N,j)\mu \end{bmatrix} \tag{4}$$

At threshold point M, the transition matrices reach steady states and become level independent. In this steady state, these matrices no longer depend on the parameter j, transitioning to steady state irreducible matrices A, B, and C respectively.
In matrix from

$$\begin{aligned} A_j &= A, \text{ for all } j \geq M \\ B_j &= B, \text{ for all } j \geq M \\ C_j &= C, \text{ for all } j \geq M \end{aligned} \tag{5}$$

For calculating various parameters, the probability of state (i, j) in the steady state is computed using the probability coefficient $P_{i,j}$ which has been introduced and defined as follows:

$$P_{i,j} = lim_{t\to\infty}[I(t) = i, J(t) = j],\ 0 \leq i \leq N, \& 0 \leq j \leq L \tag{6}$$

Where L can be finite or infinite.
The probability row vectors at the j^{th}stage are defined as

$$V_j = [P_{0,j}, P_{1,j}, ...P_{N,j}],\ \ j = 0, 1, 2... \tag{7}$$

The probability vectors, along with transient state matrices, have been represented with the help of balance equations.

$$V_0[D_0^A + D_0^B] = V_0A_0 + V_1C_1 \tag{8}$$
$$V_j[D_j^A + D_j^B + D_j^C] = V_{j-1}B_{j-1} + V_jA_j + V_{j+1}C_{j+1} \tag{9}$$
$$V_j[D^A + D^B + D^C] = V_{j-1}B + V_jA + V_{j+1}C \; M \leq j \leq L \tag{10}$$
$$V_L[D^A + D^C] = V_{L-1}B + V_LA \tag{11}$$

For an infinite state space, the balance equation is further simplified.

$$V_{j-1}B_{j-1} + V_j[A_j - D_j] + V_{j+1}C_{j+1} = 0, \; j = 0, 1...M-1 \tag{12}$$

In equation (12), $D_j = D_j^A + D_j^B + D_j^C$ where D_j^R ($R = A$ or$B$ or C) represents the diagonal matrix, whose diagonal elements are the sum of each corresponding row of the matrix R_j.

The threshold condition for the system attains at $M = N$. After reaching this threshold condition, the system enters a steady state, i.e., the system's behavior no longer depends on j. Consequently, balance equations for $j = M, M+1, \ldots$ are as follows.

$$V_{j-1}B + V_j[A - D] + V_{j+1}C = 0 \tag{13}$$

Furthermore, the total probability of the system always remains at 1, i.e.,

$$\sum_{j=0}^{\infty} V_j.e = 1 \tag{14}$$

Here $'e'$ represents the n^{th}- order column matrix with elements equal to 1.

To find the probability vectors V_j, the balance equations can be reformulated in terms of eigenvalues and eigenvectors. Equation (14) leads to a quadratic equation from which eigen values and their corresponding left eigen vectors can be derived. These values will be essential in computing performance measures.

Let's define diagonal matrices $Q_0, Q_1,$ and Q_2 with sizes $(N+1) \times (N+1)$ from the study state matrices A, B, C, as $Q_0 = B, Q_1 = A - D^A - D^B - D^C, Q_2 = C$.
Then the balance equations can be expressed in terms of quadratic form.

$$V_jQ_0 + V_{j+1}Q_1 + V_{j+2}Q_2 = 0; \quad \text{where } (M-1) \leq j \leq (L-2) \tag{15}$$

From this, the characteristic matrix polynomial further can be expressed as:

$$Q(\lambda) = Q_0 + Q_1(\lambda) + Q_2\lambda^2 \tag{16}$$

Where

$$\psi Q(\lambda) = 0; \; |Q(\lambda)| = 0. \tag{17}$$

Here λ, and ψ are the eigenvalues and left eigenvectors of the quadratic polynomial $Q(\lambda)$ respectively.

To compute eigenvectors and their corresponding eigenvalues, we further simplify the quadratic form in matrix form.

$$Q = \begin{bmatrix} 0 & -T_0 \\ I & T_1 \end{bmatrix} \tag{18}$$

where $T_0 = B/C$ and $T_1 = [A–D_A–D_B–D_C]/C$

It can be further expressed in quadratic notation form as:

$$\psi[T_0 + T_1\lambda + \lambda^2] = 0 \tag{19}$$

It can be expressed in a square matrix form of order $2N$ to compute eigen values and their corresponding eigen vectors.

$$\psi \begin{bmatrix} 0 & -T_0 \\ I & T_1 \end{bmatrix} = \lambda\psi \tag{20}$$

Since the matrix $\begin{bmatrix} 0 & -T_0 \\ I & T_1 \end{bmatrix}$ is a $2N$ size matrix, hence it has a $2N$ set of eigen values and their corresponding left eigen vectors. Out of these $2N$ eigenvalues, N eigenvalues are exactly less than one in magnitude.

Now, by considering these N eigenvalues and their corresponding half of left eigenvectors, the probability vectors can be computed as:

$$V_j = \sum_{(k=0)}^{N} a_k\psi_k\lambda_k^{j-M+1} \tag{21}$$

Inverting the next N eigenvalues that are greater than one and considering their corresponding half of left eigenvectors, the probability vector can be defined for the finite state as:

$$V_j = \sum_{(k=0)}^{N} a_k\psi_k\lambda_k^{(j-M+1)} + \sum_{k=1}^{N} b_k\phi_k\beta_k^{L-j} \tag{22}$$

Where a and b are arbitrary constants. β and ϕ are the reciprocal eigenvalues with magnitude greater than or equal to one and their left eigenvectors of the matrix Q, respectively.

Which can be further resolved in the form of state probabilities as follows:

$$p_{i,j} = \sum_{k=0}^{N} a_k\psi_k(i)\lambda_k^{j-M+1} + b_k\phi_k(i)\beta_k^{L-j} \text{ where } M-1 \le j \le L \tag{23}$$

The arbitrary constants a_k and b_k, $(k = 0, 1, \ldots, N)$ are either scalar real constants or complex values that need to be computed. These constants can be found with balance equations to compute various performance measures such as service probability, mean waiting time and loss probability of the customers, and other measures.

3. Performance Efficiency Factor

In the contemporary world, the ever-growing need for enhanced computational speed is fueled by technological advancements. One approach to achieving this goal involves partitioning computational tasks among multiple servers. This can be realized through the implementation of a coupled system or by leveraging cloud computing techniques.

While pursuing the development of a new system, it is crucial to consider the limitations of network communications and potential delays in work arising from increased computational and transmission times. Additionally, the system should be designed to be cost-effective. This section delves into the discussion of boosting speed through the utilization of multiple servers, acknowledging certain constraints and employing parallel server configurations.

The computation of the parallel computing performance factor has been considered. The speedup factor ($S_{up}(p)$) is defined to measure adequate performance by adding additional servers.

$$S_{up}(p) = \frac{t_s}{t_p} = \frac{\text{Execution time of single server}}{\text{Execution time of multiple servers}} \tag{24}$$

$$= \frac{2n^2}{2\frac{n^2}{p} + p(t_{startup} + t_{data}) + n^2(t_{startup} + 4t_{data})} \tag{25}$$

$$= \frac{n(2n+4)}{\frac{n}{p}(2n+4) + p(t_{startup} + nt_{data})} \tag{26}$$

The effect of computation and communication times play crucial role in performance of a system. As the size of the system increases, exchanging information among the servers gradually increases. The rate between these two factors can be expressed as:

$$t_{p/c} = \frac{2\frac{n^2}{p}}{p(t_{startup} + 2t_{data}) + 4n^2(t_{startup} + t_{data})} \tag{27}$$

Where $t_{p/c}$ represents the ratio processing time over communication time.

$$t_{p/c} = \frac{t_{comp}}{t_{comm}} = \frac{\frac{n}{p}(2n+4)}{pt_{startup} + nt_{data}} = O\left(\frac{n^2/p}{p+n^2}\right) \tag{28}$$

Which suggests improvement with larger n (scalable).

Several factors affect the maximum not performing speed of the system. These factors include:

- All servers are not performing effectively, and in the meantime, some servers are idle.
- Communication between processes is another factor in reducing speed.
- Effective utilization of other peripherals in multiple service systems.

Anticipating heightened customer arrivals during peak times, it is rational to allocate additional power, while reverting to standard services during non-peak periods. Initially, N servers are designated for service. When faced with peak arrivals, additional M servers are dynamically assigned to bolster power for seamless task execution. The fraction of work, denoted as f, is undertaken by the extra power servers (M servers), while the remaining fraction of work $(1-f)$ is handled by uniform servers ($N-M$ servers). Let t_s be the total time required to complete the work, p represent the single server performance speed factor, and k be the dynamic power increasing factor from servers. This graphical representation is illustrated in Figure 2, depicting the relationship between the performance efficiency factor and dynamic power allocation for a portion of the work.

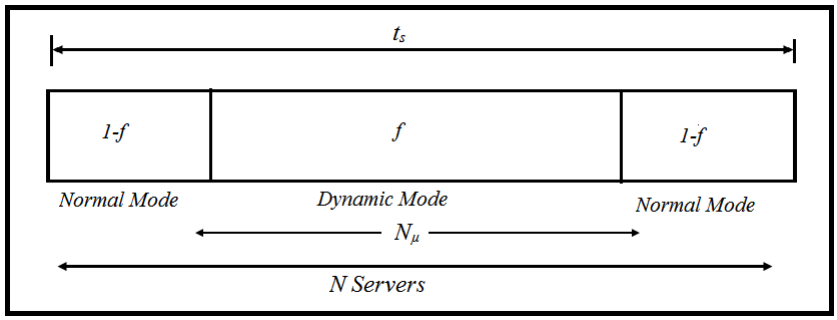


Figure 2: *Performance efficiency factor and dynamic power allocation for part of the work.*

Then the speedup performance factor represents.

$$S_{up}(p) = \frac{t_s}{ft_s\frac{M}{P}+(1-f)t_s\frac{N-M}{kp}} \tag{29}$$

$$= \frac{kp}{N-M+f[(k+1)M-N]} \tag{30}$$

From this, the following cases can be derived.

Case 1: If a customer arrives uniformly throughout the period, the service is performed typically. Hence, the fraction of work computed with dynamic mode becomes zero.

In this case, $M = 0$. Hence, the speedup performance factor becomes:

$$S_{up}(p) = \frac{kp}{N} \tag{31}$$

Case 2: If the service is expected with extra power throughout the system life, every server works dynamically. In this case, M becomes N. Hence:

$$S_{up}(p) = \frac{p}{N} \tag{32}$$

4. Result Analysis

This section presents numerical results in graphical format, utilizing various parameters to assess the performance of queuing systems when dynamic services are employed during busy periods. The comprehensive performance measures are provided for the proposed approach to compare with traditional method. The results indicate that the performance is notably improved with the proposed dynamic mode power shifting method. It demonstrates effective job handling during peak periods, while normal mode service can be seamlessly executed with uniform service during the system's general arrivals. Simulation results, depicted graphically, offer insights into the model's performance as described in the preceding section, with modifications to various parameters.

In presenting the results, certain parameters have been kept constant unless explicitly mentioned for a specific experiment. Unless otherwise it is not mentioned, the parameters are fixed as $= 1.8$, $\mu_0 = 0.1$, $\eta_N = 0.01$, $\xi_0 = 0.02$, $\xi = 0.8$, and the maximum number of servers operating in the system, $N = 10$, have been consistently maintained throughout the experiments. A substantial effort has been invested in ensuring a high degree of accuracy in this work.

Figures 3 and 4 depict a queuing system with varying service rates while maintaining a fixed number of servers and uniform service distribution. The illustrations demonstrate that as the arrival rate increases, both the mean queue length and the waiting time for service also increase in response to the influx of job arrivals. These observations suggest an exponential relationship between the mean queue length, waiting time, and the rate of job arrivals. Figure 3 specifically portrays the queue length, while Figure 4 represents the time required for service.

In Figure 5, the attention is directed towards dynamic service stations, showcasing different power factors assigned to servers spanning from station 2 to station 8. Additional powers are distributed across three distinct levels: server 2 to 5, server 2 to 6, and server 2 to 8. Through this depiction, it can be deduced that the inclusion of additional servers with dynamic service capacities results in a decrease in the average waiting time for arrivals, as visually demonstrated in the graph. Figure 6 illustrates the escalating number of customers awaiting service as the arrival throughput steadily increases. This graph showcases dynamic service stations ranging from 2 to 8, each operating at different levels of service capacity.

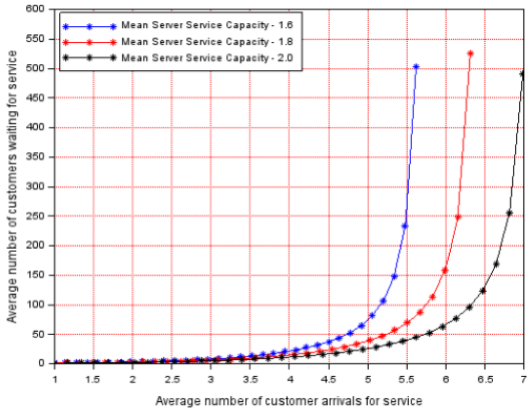


Figure 3: *Number of customers waiting for service in the queuing system over time.*

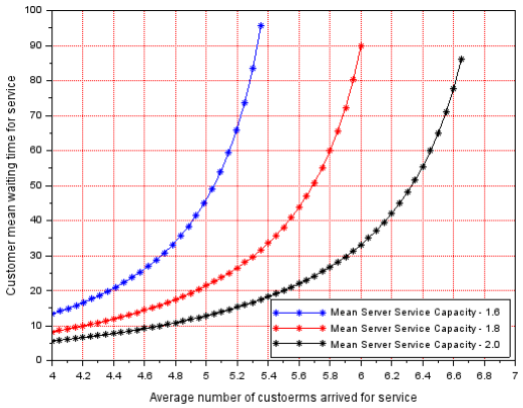


Figure 4: *Expected time of the service to the customer with mean arrival rate increases.*

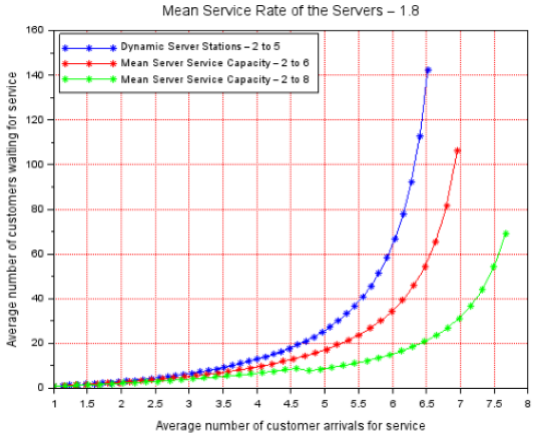


Figure 5: *Expected number of customers waiting for service with dynamic power service facility.*

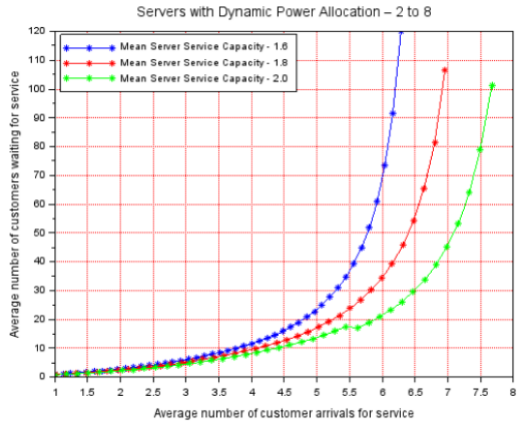


Figure 6: *The rate of increase in the number of customers varies with different levels of service capacity and arrival rates.*

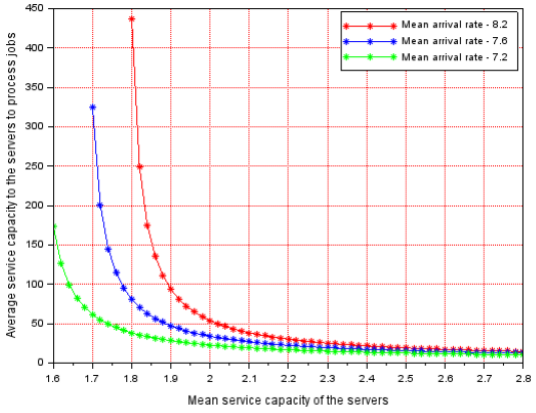


Figure 7: *The anticipated number of customers waiting for service upon joining the queue, as service capacity increases.*

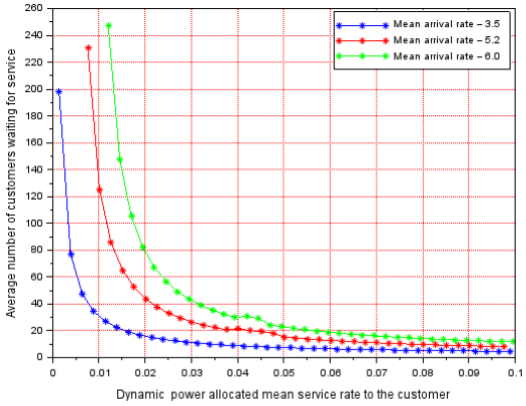


Figure 8: *Expected number of customers waiting for service as service capacity increases with server power adjustment.*

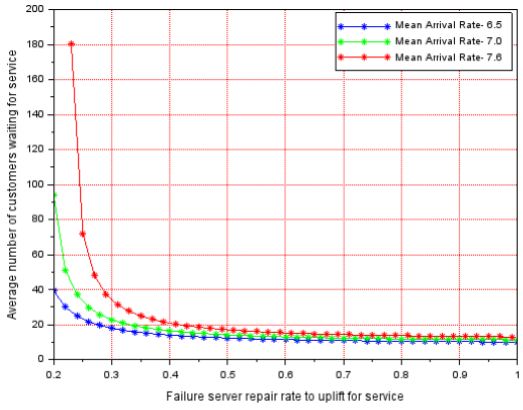


Figure 9: *Number of customers waiting for services across various arrival rates, concerning the improvement after uplifting failure server.*

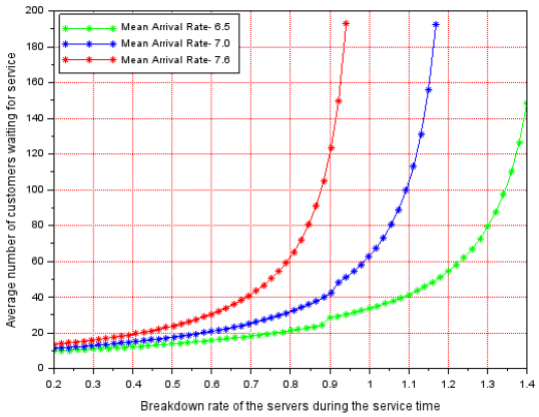


Figure 10: *Correlation between the number of customers waiting for service with servers prone to breakdowns.*

Both Figure 7 and Figure 8 depict constant arrival rates across different service capacities. In Figure 7, arrival rates of 8.2, 7.6, and 7.2 are examined within a general service system, while Figure 8 explores arrival rates of 3.5, 5.2, and 6.0 with dynamic service power shifting. The results illustrate the influence of additional power allocation to the server on the mean queue length. It's observed that as service capacity dynamically increases, there's a corresponding reduction in waiting time until it stabilizes at a constant level. This stabilization point indicates the optimal utilization level, which can be calculated from these observations. Overall, it's noted that as service capacity increases, waiting time decreases.

Referring to Figures 9 and 10, the graphs illustrate outcomes under varying arrival rates, mirroring the patterns observed in the preceding figures. In both scenarios, the machine service capacity remains constant, set with appropriate parameter values. Figure 9 reveals an intriguing observation: during periods of fast services for failure servers, service is promptly completed. Notably, an escalation in repair rate results in a significant reduction in waiting time initially, until stability is achieved. Turning to Figure 10, the findings depict the impact of varying levels

of server breakdowns on service time. It becomes evident that as the breakdown rate increases, the time taken to serve waiting customers also increases. This relationship underscores a direct proportionality between service time and failure rate.

5. Conclusion

As global customer demand continues to surge, many systems that were once efficient have become obsolete. Sustained survival requires ongoing improvements in methodologies across various real-time applications. This work addresses the need for continuous enhancement in one such application, focusing on the development of new strategies to minimize customer waiting time in queues during peak periods. The novelty of this approach lies in its dual objectives: reducing waiting time during peak hours while ensuring servers remain active during less busy periods. The validation of these methodologies with numerical values has been performed, demonstrating the practical applicability of the proposed theories in everyday scenarios.

References

[1] Akyildiz, I. F., Lee, W. Y., Vuran, M. C. and Mohanty, S. (2006). NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey. Computer networks, 50(13), 2127–2159. doi: 10.1016/j.comnet.2006.05.001

[2] Ata, B. and Shneorson, S. (2006). Dynamic control of an M/M/1 service system with adjustable arrival and service rates. Management Science, 52(11), 1778–1791. doi: 10.1287/mnsc.1060.0587

[3] Bouchentouf, A. A., Guendouzi, A. and Majid, S. (2020). On impatience in Markovian M/M/1/N/DWV queue with vacation interruption. Croatian Operational Research Review, 11(1), 21–37. doi: 10.17535/crorr.2020.0003

[4] Chakka, R. (1995). Performance and reliability modelling of computing systems using spectral expansion. Doctoral dissertation, Newcastle University. Retrieved from: theses.ncl.ac.uk/jspui/handle/10443/2112

[5] Chakka, R. and Van Do, T. (2007). The MM $\Sigma_{k=1}^{K}$CPPk/GE/c/L G-queue with heterogeneous servers: Steady state solution and an application to performance evaluation. Performance Evaluation, 64(3), 191–209. doi: 10.1016/j.peva.2006.05.001

[6] Chan, C. W., Huang, M. and Sarhangian, V. (2021). Dynamic server assignment in multiclass queues with shifts, with applications to nurse staffing in emergency departments. Operations Research, 69(6), 1936–1959. doi: 10.1287/opre.2020.2050

[7] Chen, D. and Trivedi, K. S. (2005). Optimization for condition-based maintenance with semi-Markov decision process. Reliability engineering and system safety, 90(1), 25–29. doi: 10.1016/j.ress.2004.11.001

[8] Devi, K. L. and Valli, S. (2021). Multi-objective heuristics algorithm for dynamic resource scheduling in the cloud computing environment. The Journal of Supercomputing, 77(8), 8252–8280. doi: 10.1007/s11227-020-03606-2

[9] Diamantoulakis, P. D., Kapinas, V. M. and Karagiannidis, G. K. (2015). Big data analytics for dynamic energy management in smart grids. Big Data Research, 2(3), 94–101. doi: 10.1016/j.bdr.2015.03.003

[10] Elliriki, M., Reddy, C. S., Anand, K. and Saritha, S. (2022). Multi server queuing system with crashes and alternative repair strategies. Communications in Statistics-Theory and Methods, 51(23), 8173–8185. doi: 10.1080/03610926.2021.1889603

[11] Gandhi, A., Harchol-Balter, M., Das, R. and Lefurgy, C. (2009). Optimal power allocation in server farms. ACM SIGMETRICS Performance Evaluation Review, 37(1), 157–168. doi: 10.1145/2492101.1555368

[12] Harrison, J. M. and Zeevi, A. (2004). Dynamic scheduling of a multiclass queue in the Halfin-Whitt heavy traffic regime. Operations Research, 52(2), 243–257. doi: 10.1287/opre.1030.0084

[13] Mamatha, E., Sasritha, S. and Reddy, C. S. (2017). Expert system and heuristics algorithm for cloud resource scheduling. Romanian Statistical Review, 65(1), 3–18.

[14] Mamatha, E., Saritha, S., Reddy, C. S. and Rajadurai, P. (2020). Mathematical modelling and performance analysis of single server queuing system-eigenspectrum. International Journal of Mathematics in Operational Research, 16(4), 455–468. doi: 10.1504/IJMOR.2020.108408

[15] Moridi, E., Haghparast, M., Hosseinzadeh, M. and Jassbi, S. J. (2020). Fault management frameworks in wireless sensor networks: A survey. Computer communications, 155, 205–226. doi: 10.1016/j.comcom.2020.03.011

[16] Sadana, U., Chenreddy, A., Delage, E., Forel, A., Frejinger, E. and Vidal, T. (2024). A survey of contextual optimization methods for decision-making under uncertainty. European Journal of Operational Research. doi: 10.1016/j.ejor.2024.03.020

[17] Saritha, S., Mamatha, E. and Reddy, C. S. (2019). Performance Measures of Online Warehouse Service System with Replenishment Policy. Journal Europeen Des Systemes Automatises, 52(6), 631–638. doi: 10.18280/jesa.520611

[18] Saritha, S., Mamatha, E., Reddy, C. S. and Anand, K. (2019). A model for compound poisson process queuing system with batch arrivals and services. Journal Europeen des Systemes Automatises, 53(1), 81–86. doi: 10.18280/jesa.530110

[19] Saritha, S., Mamatha, E., Reddy, C. S. and Rajadurai, P. (2022). A model for overflow queuing network with two-station heterogeneous system. International Journal of Process Management and Benchmarking, 12(2), 147–158. doi: 10.1504/IJPMB.2022.121592

[20] Shirdel, G. H. and Abdolhosseinzadeh, M. (2016). The critical node problem in stochastic networks with discrete-time Markov chain. Croatian Operational Research Review, 7(1), 33–46. doi: 10.17535/crorr.2016.0003

[21] Sreelatha, V., Mamatha, E., Anand, S. K. and Reddy, N. H. (2022). Markov Process Based IoT Model for Road Traffic Prediction. In International Conference on Modeling, Simulation and Optimization, 329–338. doi: 10.1007/978-981-99-6866-4_24

[22] Sreelatha, V., Mamatha, E., Reddy, C. S. and Rajdurai, P. S. (2022). Spectrum relay performance of cognitive radio between users with random arrivals and departures. In Mobile Radio Communications and 5G Networks: Proceedings of Second MRCN 2021, 533–542. doi: 10.1007/978-981-16-7018-3_40

[23] Yang, T., Zhao, L., Li, W. and Zomaya, A. Y. (2021). Dynamic energy dispatch strategy for integrated energy system based on improved deep reinforcement learning. Energy, 235, 121377. doi: 10.1016/j.energy.2021.121377

Croatian Operational Research Review
CRORR **15:2**(2024), 185–198

A hybrid population-based algorithm for solving the Minimum Dominating Set Problem

Belkacem Zouilekh[1] **and Sadek Bouroubi**[1,*]

[1] *LIFORCE Laboratory, Department of Operations Research, Faculty of Mathematics, University of Sciences and Technology Houari Boumediene, P.B. 32 El-Alia, 16111, Bab Ezzouar, Algiers, Algeria.*
E-mail: ⟨{bzouilekh, sbouroubi}@usthb.dz⟩

Abstract. The Minimum Dominating Sets (MDS) problem is pivotal across diverse fields like social networks and ad hoc wireless networks, representing a significant NP-complete challenge in graph theory. To tackle this, we propose a novel hybrid cuckoo search approach. While cuckoo search is renowned for its wide search space exploration, we enhance it with intensification methods and genetic crossover operators for improved performance. Comprehensive experimental evaluations against state-of-the-art techniques validate our approach's effectiveness.

Keywords: genetic algorithm, hybrid cuckoo search algorithm, minimum dominating set

Received: March 16, 2024; accepted: June 5, 2024; available online: October 7, 2024

DOI: 10.17535/crorr.2024.0015

1. Introduction

Given an undirected graph $G = (V, E)$, a subset S of V is named a dominating set of G if each vertex $v \in V$ is either an element of S or is adjacent to an element of S. Such a subset is called a dominating set of G, and we say S dominates G or G is dominated by S. The minimum cardinality of a dominating set in G is denoted by $\gamma(G)$. If S dominates G, such as $|S| = \gamma(G)$, then S is called a Minimum Dominating Set, or MDS for short [10]. As an example, Figure 1 shows a subgraph of Twitter tweets with 85 vertices that spread specific 5G false news that is linked directly to the COVID-19 pandemic [20], with nodes denoting Twitter users and edges representing follower relationships. The MDS obtained by our approach is highlighted in red. The original status of the publisher is represented by the vertices of the dominating set, which are marked in red. If we assume the simplest scenario: followers, marked in black, cannot retweet and content cannot spread out of followers, only 2 users could probably spread misinformation among the other 83 users.

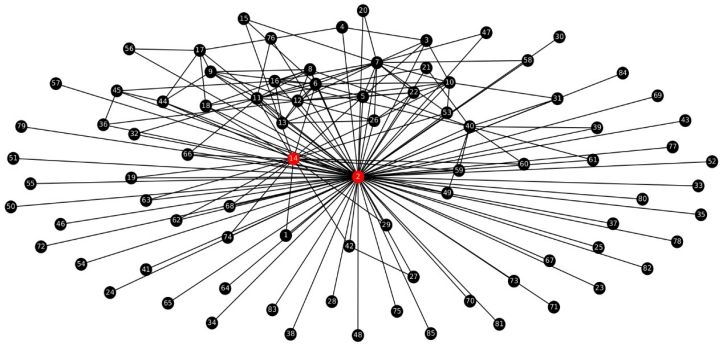


Figure 1: *Representations of a real-induced sub-graph from Twitter.*

*Corresponding author.

The minimum dominating sets and their variations, such as the minimum connected dominating set and the minimum weighted dominating set, have pervasive applications in a wide range of disciplines, besides routing in ad hoc wireless networks [26], sensor networks, and MANETs [3]. In addition, the MDS can be employed to determine the main subject sentences for document summary via sentence network design [27]. Furthermore, the MDS may be used to model and investigate the positive impact of social networks [7]. Moreover, controlling the spread of epidemics [22] also involves early diagnosis and control of epidemic spreading in various areas of human society, such as virus spread in computer systems, misinformation (as shown in Figure1), and content diffusion via social media [30]. In this paper, we address the minimum dominating set problem by proposing a new Hybrid Cuckoo Search Algorithm, henceforth called HCSA-MDS, that combines the cuckoo search algorithm and the genetic crossover operator with intensification schemes as well as repairing and cleaning to achieve effective exploration and exploitation.

1.1. Paper structure

The remainder of this paper is partitioned into many sections. In Section 2 that follows, we present some linked works, including exact algorithms and heuristics, applied to solving the MDS problem. Section 3 goes into detail about the Cuckoo Search Algorithm. The hybrid CSA algorithm is then presented and described in Section 4. Section 5 summarizes the results of the computation. The paper is finally concluded in Section 6.

2. Related works

The MDS problem is an NP-Hard combinatorial problem [8], and it has been thoroughly investigated using exact (exponential-time) techniques. The best known exact algorithm for the MDS problem performs in $\mathcal{O}(1.4864^n)$ time and polynomial space, constructed through the measure and conquer approach by Y. Iwata [14]. Unfortunately, the exact techniques that execute at an exponential scale are only possible for limited-size networks, severely limiting their effective uses. As a result, scientific researchers have mainly concentrated on stochastic computational heuristics and, lately, metaheuristics. Hence, many heuristic approaches have been adopted in the state of the art to handle the MDS problem, such as[16, 25, 2, 15]. Moreover, L. A. Sanchis [19] performed experimental research on different heuristic approaches in this perspective; he thoroughly investigated many greedy methods for the MDS problem. After that, Ho et al. [13] presented ACO-TS, an improved Ant Colony Optimization metaheuristic that integrates a technique for stimulating the building of different solutions based on a concept adopted from genetic algorithms called tournament selection. Subsequently, in [11], Hedar and Ismail presented a hybrid genetic algorithm referred to as HGA-MDS that employs a local search characterized by three intensification techniques. Furthermore, in [12], Hedar and Ismail also suggested a SAMDS metaheuristic addressing the MDS problem by employing a Stochastic Local Search (SLS) algorithm for strengthening a solution by looking for a stronger one in its near area. The SLS is enhanced by using a simulated annealing process. Recently, Abed and Rais [1] introduced a hybrid population-based technique known as the Hybrid Bat Algorithm, which is rooted in microbat bio-sonar characteristics and simulated annealing. The fact that SA is efficient in exploitation and the bat algorithm has a high capacity for the exploration of large regions in the search space helps to ensure a good balance between intensification and diversification in the search methodology.

3. Cuckoo Search Algorithm

The Cuckoo Search Algorithm (CSA), based on the fascinating breeding behavior of particular cuckoo species, such as brood parasitism, is one of the most recently developed metaheuristics

[9]. To describe the Cuckoo Search for clarity, Yang and Deb use the following three idealized principles [28]:

1. One egg is laid by each cuckoo at a time, and it deposits it in a nest that is selected at random.
2. The best nests with the highest quality eggs (solutions with the highest fitness) will be passed on to future generations.
3. The number of host nests is fixed, and a host has a probability of discovering an alien egg of $P_a \in [0, 1]$. In this scenario, the host bird can either discard the egg or leave the nest and create a new one at a different site.

A solution's fitness can simply be proportional to the objective function's value. A cuckoo egg signifies a new solution, and each egg in a nest indicates a solution. The goal is to replace the poor solutions in the nests with new and maybe improved ones (cuckoos). This approach can be expanded to a more sophisticated scenario of several eggs representing a set of solutions in each nest. Therefore, we will take the simplest approach possible, with each nest containing only one egg. The CSA pseudo-code is shown in Algorithm 1 based on these criteria.

Algorithm 1: Cuckoo Search Algorithm: CSA

Input: Problem instance s
Output: The best possible solution s^*
Initialization:
Initialize the population of m host nests (solutions) $s = (s_1, \ldots, s_m)$;,
maxGen : Maximum number of generations,
t $\leftarrow 0$.
while $t \leq maxGen$ **do**
  Get a cuckoo (say x_i) at random using Lévy flights and evaluate its fitness $f(x_i)$;
  Choose one of n (say y_i) nests at random.
  if $f(x_i) > f(y_i)$ **then**
    Replace the old nest x_i by the new one y_i;
  else
    Continue
  A portion of the worst nests p_a are removed and replaced with new ones.
  Sort the solutions and choose the best one.
  Refresh the current best solution.
  $t = t + 1$
return s^*

After generating the initial population, CSA generates new solutions x^{i+1} associated to each cuckoo i in each iteration t by a random walk via Lévy flight:

$$x_i^{t+1} = x_i^t + \alpha \oplus L\acute{e}vy(\lambda) \tag{1}$$

Lévy flights are a type of random walk named after the French mathematician Paul Lévy [5], in which α above in the equation denotes the step size, which should correspond to the problem's interests (most of the time $\alpha = 1$), and the term product $\oplus$ denotes entrywise multiplications. The step lengths are selected from a probability distribution with a power law tail. Lévy distributions, or stable distributions, are the names given to these probability distributions.

$$L\acute{e}vy \sim u = l^{\lambda}, \tag{2}$$

where $1 < \lambda \leq 3$ and l is the flight length.

The initial purpose of CSA and Lévy flights was to solve continuous optimization problems. Yang and Deb [28] show the outperformance and robustness of CSA over GA and PSO because of its larger search space exploration capacity and fewer parameters to fine-tune than other algorithms. Actually, there is just one parameter P_a, separate from the population size N. CSA has been widely used and shown promising efficiency in a variety of optimization and computational intelligence applications [29]. On the other hand Boumedine and Bouroubi proposed a discrete hybrid cuckoo search algorithm to solve the protein folding problem [4]. In this paper, we present a discrete hybrid CS-based method, called Hybrid Cuckoo Search Algorithm for Minimum Dominating Set (HCSA-MDS), to solve the MDS problem, which is one of the most difficult combinatorial optimization problems. The suggested discrete HCSA-MDS employs an adaptive Lévy flight for the MDS problem.

4. Hybrid CSA for the MDS problem

We describe our approach to tackling the MDS problem in the following section. Our algorithm takes as input a problem instance $G = (V, E)$, in which G is an undirected, connected graph, V is a set of vertices, and E is a set of edges. The HCSA-MDS pseudo-code shown in Algorithm 4 combines the cuckoo search algorithm with the genetic crossover operator and the cleaning with repair technique to exploit the solutions. Lévy flight, with its ability to explore new regions in the search space, is a particularly useful strategy in the diversification phase. The goal of this hybridization is to establish a proper balance between exploration and exploitation. The main structure is shown in Figure 2.

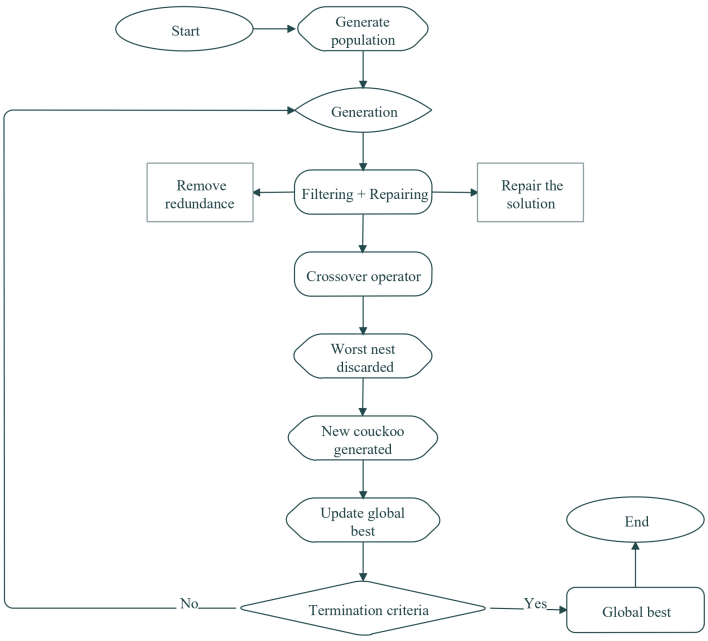


Figure 2: *HCSA-MDS flowchart.*

We first discuss the HCSA-MDS components before fully stating pseudo-code 4.

4.1. Solution and population representation

The HCSA-MDS algorithm starts with a population P that contains a set of m nests (solutions), some of which are non-feasible. As a result, a $0-1$ vector with a dimension equal to the number

of nodes in the graph $G = (V, E)$ represents a solution x in P. The i-th node in G is part of the dominating set if the vector's i-th element has a value of 1. While the i-th vector's item has a value of 0, this indicates that a node is not part of the dominant set. Table 1 illustrates the binary representation of the solution to Example 1 presented in Section 1.

1	2	3	$\cdots$	13	14	15	$\cdots$	85
0	1	0	$\cdots$	0	1	0	$\cdots$	0

Table 1: *Binary representation of the best found MDS.*

4.2. A new cuckoo egg's production using the crossover operator

The crossover operator is one of the genetic operators that applies to two parents (solutions) and then chooses at random any of the crossover points p_h, $h \in \{0, 1, ..., n-1\}$ [23]. Two offspring (new solutions) are created by joining the parents at the crossover point for the next generation. However, for any solution from population P that we consider the first parent, we choose a second parent in a random way from the current population. Then, the one point crossover operation is used to generate two offspring. In the proposed algorithm, only the best offspring is chosen for the next stage; see the example in the following Table 2:

Parent	Binary representation		Offspring	Binary representation
First	1010|10110	$\Rightarrow$	First	1010|10010
Second	1111|10010		Second	1111|10110

Table 2: *An illustrative example of a crossover operator with a single point-crossover.*

In the previous example, two offspring were produced randomly by combining the first and second parents at the fourth position. The crossover operation in this example results in two novel solutions, where the first offspring exhibits superior performance in the minimum dominating set problem compared to both its parents and sibling.

4.3. Filtering and Reparation procedures

HCSA-MDS employs a filtering mechanism to enhance the MDS solutions by removing redundant nodes, thereby limiting the size of the dominating set while preserving its coverage [11]. This filtering procedure is detailed in Algorithm 2. Complementing this, the reparation procedure addresses solutions S that fail to cover all nodes. It begins by verifying if the solution is not a dominating set. Subsequently, the next vertex added to the set of dominators is chosen from the non-dominating vertices with the highest degree. This reparation process continues until S constitutes a valid dominating set, as illustrated in Algorithm 3.

Algorithm 2: Filtering procedure

Input: Solution S.
Output: Filtered solution.

```
for each node x in S do
    if x = 1 then
        Set x = 0
        Compute the new fitness value of the solution S
        if the new fitness value is increased then
            Update the solution S and set x = 0
        else
            Reset x = 1
return S
```

Algorithm 3: Repairing procedure

Input: Solution S.
Output: Repaired solution.

```
if There are non-covered nodes in the solution S then
    Select a node x with maximum degree in non-covered nodes set;
    Set x = 1 and update the solution S;
    Remove x from non-covered nodes and its neighbours;
return S
```

4.4. Construction of a new solution via Lévy flights

Due to its unique step-length characteristics, the Lévy flight strategy efficiently explores search spaces. To apply this technique to the Minimum Dominating Set (MDS) problem, we relate the step length, representing the length of the subset, to the values generated by Lévy flights. For instance, if S S is an n-dimensional solution, we partition the interval $[0, 1]$ into m subintervals and define the step length ($step_{length}$) as follows:

θ	$\left[0, \frac{1}{m}\right]$	$\left[\frac{1}{m}, \frac{2}{m}\right]$	...	$\left[\frac{m-1}{m}, 1\right]$
$step_{length}$	$\left[1, \frac{\max_l}{m}\right]$	$\left[\frac{\max_l}{m}, \frac{2\max_l}{m}\right]$	...	$\left[\frac{(m-1)\max_l}{m}, \max_l\right]$

where θ is the Lévy flight value acquired and $\max_l = \frac{n}{h}$, $h \in \{1, 2, 3, \ldots, n\}$ is the maximum subset we are able to invert. Let i be the length of the subset chosen at random from the interval generated by Lévy's flight value. Then, from the n-dimensional solution, we choose a random position between 1 and $n - i$, from which the i-inversion begins.

5. Experimental results

HCSA-MDS effectiveness is evaluated against a collection of well-known efficient algorithms for the Minimum Dominating Set problem that have been reported in the literature. The rest of this section will be as follows: First, all benchmarks used are presented. Then, we compare the results obtained by the proposed algorithm with state-of-the-art approaches. We use Wilcoxon's signed-rank test to compare the HCSA-MDS method with other approaches. This nonparametric test is suitable for matched-pair data and detects substantial performance differences. We also use Critical Difference (CD) diagrams to visualize results, as described in [6]. First, a Friedman test checks for significant performance differences among algorithms. If significant, post-hoc analysis follows. CD diagrams rank algorithms on a horizontal axis, with the best near 1. Statistically similar algorithms are connected by bars. For more details, visit this link: https://mirkobunse.github.io/CriticalDifferenceDiagrams.jl/stable.

5.1. Data Sets

The Minimum Dominating Set (MDS) problem commonly uses two benchmark datasets. The first includes 42 random geometric graphs with up to 400 nodes, as outlined in [12]. These networks are generated by randomly placing n nodes within an M × M area, following specific instructions detailed in [13]. Multiple graph instances are created per network based on specified range parameters. The second dataset comprises 20 graphs with 400 and 800 vertices sourced from [12], where optimal dominating sets are known. These graphs are generated using a methodology detailed in [19]. For additional benchmark details, please refer to Table 3.

Network Id	N° of nodes (n)	Range	Area (A)	N° of instances
$N1_A^n$	80	60-120	400×400	7
$N2_A^n$	100	80-120	600×600	5
$N3_A^n$	200	70-120	700×700	6
$N4_A^n$	200	100-160	1000×1000	7
$N5_A^n$	250	130-160	1500×1500	4
$N6_A^n$	300	180-220	2000×2000	5
$N7_A^n$	350	200-230	2500×2500	4
$N8_A^n$	400	210-240	3000×3000	4

(a) *First benchmark.*

Network	n	p	d	N° of instances
$N_{d,0.1}^{400}$	400	0.1	$8, 11, 14, 18, 23$	5
$N_{d,0.3}^{400}$	400	0.3	$3, 5, 8, 11, 14$	5
$N_{d,0.5}^{400}$	400	0.5	$3, 5, 8, 11$	4
$N_{d,0.1}^{800}$	800	0.1	$11, 14, 22$	3
$N_{d,0.3}^{800}$	800	0.3	$3, 5$	2
$N_{d,0.5}^{800}$	800	0.5	$3, 6$	2

(b) *Second benchmark.*

Table 3: *Benchmark data sets.*

5.2. Tuning algorithm parameters

To conduct our experiment, we set the cuckoo search parameters according to the commonly used values in the literature for various optimization problems [28]. For a fair comparison, we selected the population size N and the number of generations $maxGen$ based on previous works addressing similar problem. The population size N was set to 40, the number of generations $maxGen$ to 100, the probability of the host bird discovering a cuckoo egg Pa to 0.25, and the cuckoo's step length parameters γ and α to 1.5 and 1 respectively.

Algorithm 4: HCSA-MDS Algorithm

Input: A graph $G = (V, E)$.
Output: Dominating Set.
Initialization:
Generate an initial population P of m feasible solutions, $S = (x_i, \ldots, x_m)$,
maxGen : Maximum number of generations,
Gen $\leftarrow 0$.

```
while Gen ≤ maxGen do
    i = 1
    while i ≤ m do
        Filter the repaired solution to improve its quality
        Select a random solution x_j from P as second parent
        Produce a new cuckoo egg y_j using the crossover operator for the selected
          parents x_i and x_j
        if f(y_j) ≥ f(x_i) then
            Replace x_i by the new produced solution y_j
        i = i + 1
    Gen = Gen + 1
The worst solutions are removed from P, and new ones are generated via Lévy flight
  proportional to P_a ∈ [0, 1]
Rank the solutions from the best to the worst and find the best one
Update the global optimal solution
return Dominating Set
```

5.3. The numerical performance

Each benchmark was performed 20 times on the first benchmark and 10 times on the second to ensure a fair comparison. Our HCSA-MDS algorithm was implemented in Python. All testing was conducted on a system with 16 GB of RAM and a 2.6 GHz Intel Core i5 CPU. The source

code and datasets for reproducing the experiments are available at https://github.com/elkacem/HCSA_MDS .

5.3.1. Comparative analysis of HCSA-MDS against the state-of-the-art approaches on the first benchmark

We test the suggested HCSA-MDS on the first benchmark against HGA-MDS [11], SAMDS [12], and HBA [1] for the purpose of proving the performance of the HCSA-MDS approach. Check Table 4. The performance of HCSA-MDS is evaluated using the following metrics for each instance:

1. **Best**: The smallest dominating set found across all runs for each graph instance.
2. **Avg.(Std.)**: The average and standard deviation of the best solution values from individual runs for each graph instance.
3. **Worst**: The worst solution found in 10 runs for each graph instance (only for HCSA-MDS).
4. **Rea.**: The number of runs where the optimal solution (domination number) was achieved.

The following assessment can be drawn from the previous experiments (see Table 4):

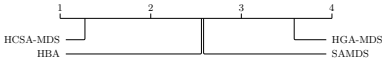
• HCSA-MDS is able to improve the best-known solution in 27 out of 42 graph instances and match the best-known solution for the rest of the 14 instances. And only in one instance ($N5_{1500}^{250}$, 160) did HBA provide a better solution. For the larger graphs, the differences between HCSA-MDS and the other algorithms begin to grow. For example, in instances with 300, 350, and 400 vertices, HCSA-MDS provided better solutions in all graph instances, with a substantial difference, as shown in Figure 4, which simulates the results of a best-solution comparison between HCSA-MDS and the other approaches. In addition, we used the critical difference diagram in Figure 3a to verify these results, showing that HCSA-MDS beats all other algorithms, followed by HBA, SAMDS, and HGA-MDS, which is the worst. Summarizing, we can assert that HCSA-MDS beats the state of the art in terms of solution quality by a large margin.

• Concerning the worst solution obtained over 20 runs, in 17 out of 42 graph instances, HCSA-MDS is able to outperform the currently best solution produced in the state-of-the-art approaches. Furthermore, HCSA-MDS's worst solution matches the best solution provided by other approaches in 19 instances, and in only 6 cases, the best solution provided by the other approaches is better than the worst solution by HCSA-MDS. Table 6 displays Wilcoxon's test statistics, which show that there is a significant difference between HCSA-MDS and other approaches. Finally, Figure 3b shows the critical difference plot, where clearly HCSA-MDS exceeds the state of the art in terms of the worst solution when compared to the best solution yielded by literature approaches.

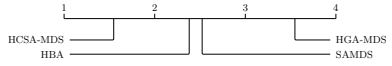
• The results of the standard deviation Table 4 demonstrates that HCSA-MDS outperforms HBA in terms of stability, as it produced stable dominating set values in 12 out of 42 instances, whereas HBA only provided stable values in 6 out of 42 instances. Additionally, the standard deviation for HCSA-MDS in the other cases is negligible, indicating high stability. In comparison, SAMDS performs better than HGA-MDS, which is the least effective, particularly in dense graphs. These findings are supported by the mean ranks presented in the critical difference plots shown in Figure 3c, which confirm that HCSA-MDS is statistically superior to the other algorithms.

Graph Id		HGA-MDS		SAMDS		HBA		HCSA-MDS		
Range		Best	Avg.(Std.)	Best	Avg.(Std.)	Best	Avg.(Std.)	Best	Avg.(Std.)	Worst
$N1_{400}^{80}$	60	**15**	15.95(0.39)	**15**	16.35(0.67)	16	16(0)	**15**	15(0)	**15**
	70	13	14(0.73)	**12**	14.40(1.14)	13	13.30(0.48)	**12**	12(0)	**12**
	80	10	10.85(0.59)	10	12(1.03)	10	10.60(0.52)	**9**	9(0)	**9**
	90	**8**	8.40(0.50)	**8**	9.60(1.27)	9	9.60(0.52)	**8**	8(0)	**8**
	100	**7**	8.20(0.52)	**7**	9.10(0.97)	8	8.40(0.52)	**7**	7(0)	**7**
	110	**6**	6.05(0.22)	**6**	7.55(0.69)	**6**	6.90(0.32)	**6**	6(0)	**6**
	120	**5**	5.95(0.39)	**5**	7(1.08)	6	6.70(0.48)	**5**	5.10(0.30)	6
$N2_{600}^{100}$	80	19	19.55(0.85)	**18**	20.55(1.23)	19	19(0)	**18**	18(0)	**18**
	90	16	17.20(0.95)	**15**	17.65(1.73)	18	18(0)	**15**	15(0)	**15**
	100	14	14.85(0.49)	**13**	14.95(1.05)	14	14.40(0.52)	**13**	13.15(0.36)	14
	110	**11**	12.15(0.67)	**11**	12.85(1.23)	12	12.30(0.48)	**11**	11(0)	**11**
	120	10	10.15(0.37)	**9**	12(1.49)	11	11(0)	**9**	9.25(0.43)	10
$N3_{700}^{200}$	70	37	45.65(10.83)	35	39.95(2.09)	34	35.40(0.84)	**32**	32.45(0.59)	34
	80	31	33.05(1.32)	29	33.50(1.73)	28	28.20(0.42)	**26**	27.45(0.59)	28
	90	26	28.75(2.67)	25	29, 25(1.94)	25	26(0.67)	**23**	23.50(0.50)	24
	100	21	24.70(4.94)	20	24.20(1.70)	21	21.80(0.42)	**18**	18.15(0.36)	19
	110	19	19.95(0.51)	18	21.40(1.93)	17	17.70(0.48)	**16**	16.65(0.48)	17
	120	17	17.30(0.47)	15	19.55(1.43)	15	16.20(0.79)	**14**	14(0)	**14**
$N4_{1000}^{200}$	100	39	45.25(5.57)	38	41.35(1.87)	36	36.70(0.48)	**35**	36.15(1.01)	38
	110	35	37.35(2.06)	33	37.20(2.44)	**30**	30.80(0.42)	**30**	30.35(0.48)	31
	120	27	28.90(1.77)	26	30.10(1.45)	26	26.30(0.48)	**23**	23.30(0.46)	24
	130	26	27.30(1.13)	25	28.15(1.42)	**23**	24.20(0.63)	**23**	23(0.59)	**23**
	140	23	24.35(1.35)	22	25.70(1.84)	22	23(0.67)	**20**	20.95(0.59)	22
	150	21	21.45(0.76)	20	23.55(1.43)	19	20.30(0.82)	**17**	17.95(0.74)	19
	160	20	21.60(0.94)	19	21.30(1.30)	18	18(0)	**17**	17(0)	**17**
$N5_{1500}^{250}$	130	55	75.45(24.01)	51	56.05(2.68)	49	50(0.67)	**47**	47.20(0.40)	48
	140	48	59.75(12.78)	46	48.65(1.35)	42	42.60(0.70)	**40**	40.85(0.36)	41
	150	44	48.30(3.34)	41	44.75(1.71)	**37**	37.90(0.32)	**37**	37.50(0.50)	38
	160	38	41.65(3.42)	37	40.90(1.92)	**31**	32.60(1.71)	34	34.15(0.36)	35
$N6_{2000}^{300}$	180	54	61.20(6.64)	47	52.35(2.41)	44	45.70(0.95)	**43**	43.45(0.50)	44
	190	48	55.55(6.35)	46	50.20(2.24)	44	44.40(0.52)	**40**	41.25(0.83)	43
	200	41	47.90(5.07)	40	45.25(2.55)	41	41.10(0.32)	**36**	36.50(0.67)	38
	210	40	48.60(7.99)	39	43.60(1.70)	37	37.70(0.82)	**34**	34.40(0.58)	36
	220	36	39.90(4.34)	36	40.65(1.90)	33	34.10(0.57)	**31**	31.70(0.46)	32
$N7_{2500}^{350}$	200	67	93.45(23.58)	61	66.35(2.13)	58	58.90(0.32)	**54**	55.85(0.85)	58
	210	63	91.20(26.70)	58	61.85(2.18)	53	55(1.05)	**48**	49.70(0.78)	51
	220	55	76.85(30.55)	49	55.05(2.31)	51	51.80(0.63)	**44**	45.25(0.70)	47
	230	51	67(21.02)	48	54.05(2.42)	45	47.10(1.37)	**43**	43.90(0.83)	45
$N8_{3000}^{400}$	210	79	115.55(41.21)	75	80.15(3.10)	72	73.30(0.95)	**68**	69.45(0.74)	71
	220	77	110.45(39.76)	73	79.25(3.18)	70	71(0.82)	**63**	64.50(0.92)	66
	230	73	111.55(38.94)	71	74.10(2.22)	64	64(0)	**60**	60.85(0.57)	62
	240	70	103.15(32.02)	63	68.80(2.98)	58	59.30(0.67)	**56**	56.95(0.74)	58

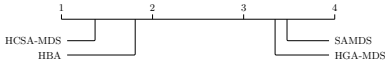
Table 4: *Performance comparison of various algorithms for the first benchmark sets.*



(a) Comparison of the best solutions (**Best**).



(b) HCSA-MDS worst solutions (**Worst**) compared to HBA,SAMDS, and HGA-MDS best solutions.



(c) Standard deviation (**std.**)

Figure 3: *Critical Difference plots evaluating HCSA-MDS, HBA, SAMDS, and HGA-MDS for the first benchmark.*

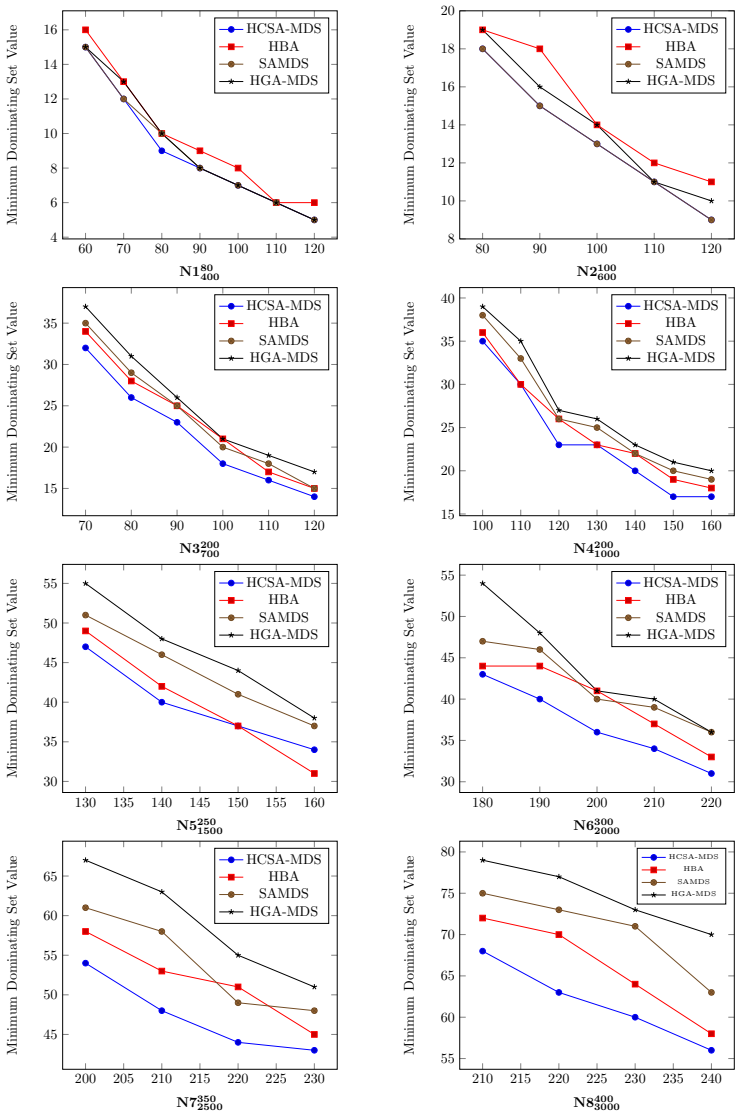


Figure 4: *Best solution analysis of various approaches.*

5.3.2. Numerical results for the HCSA-MDS and various approaches on the second benchmark

We discuss the effectiveness of the suggested HCSA-MDS on the second test case benchmark. Our method's results are evaluated against those of state-of-the-art algorithms. Hence, we compare HCSA-MDS against a variety of greedy algorithms from [19], where Sanshis L described and studied many greedy heuristics designated Greedy, GreedyRev, GreedyRan, GreedyVote, and GreedyVoteGr, which is an updated variant of GreedyVote. According to the evaluations, GreedyVoteGr beats other greedy heuristics when employing the local search process. Similarly, we compare HCSA-MDS with the Hybrid Genetic Algorithm abbreviated HGA-MDS from [11], as well as Simulated Annealing (SAMDS) from the paper [12], which shows better performance than HGA-MDS and the greedy heuristics. The numerical values are presented in Table 5. For each graph instance, the results of various approaches are presented in terms of the average solution (**Avg.**) obtained in 10 runs in each table and how many runs out of 10, the Minimum Dominating Set was reached, which is represented in the column labeled "**Rea.**". Three alternative graph densities are taken: from dense to sparse, 0.5, 0.3, and 0.1. The experiments in Table

Criteria	Algorithm	Z	Sig. (p value)
Optimal Reached (Opt.)	HGA-MDS ***vs.*** GrVoteGr	-3.306	0.001
	SAMDS ***vs.*** HGA-MDS	-2.364	0.018
	HCSA-MDS ***vs.*** HGA-MDS	-3.026	0.002
	HCSA-MDS ***vs.*** SAMDS	-2.555	0.011
Average (Avg.)	HGA-MDS ***vs.*** GrVoteGr	-3.351	0.001
	SAMDS ***vs.*** HGA-MDS	-1.601	0.109
	HCSA-MDS ***vs.*** HGA-MDS	-3.051	0.002
	HCSA-MDS ***vs.*** SAMDS	-2.677	0.007
Worst of HCSA-MDS VS Best of Others	HCSA-MDS vs HBA	-3.304	0
	HCSA-MDS vs SAMDS	-4.634	0
	HCSA-MDS vs HGA-MDS	-5.108	0

Table 6: *Comparison of statistical test results for different algorithms.*

5 shows that HCSA-MDS is the highest-performing algorithm, with absolute distinctions in terms of average solution and reaching the optimal solution. Moreover,almost all instances reached the optimal in all 10 runs except two instances ($\mathbf{N^{400}_{0.3,3}}$,$\mathbf{N^{800}_{0.5,3}}$3) optimal reached 8, 9 times respectively out of 10. To determine the relevance of the differences between the approaches results and HCSA-MDS, we used Wilcoxon's signed-rank test. It only makes sense to consider only the GreedyVoteGr, HGA-MDS, and SAMDS approaches, as they outperform the other greedy algorithms; moreover, the results of the rest of the greedy methods are too similar. The test statistics presented in Table 6 show that there is a significant difference between HGA-MDS and the best-performing algorithm in the Greedy series from [19], which is GreedyVoteGr, in terms of both criteria. Furthermore, HGA-MDS could perform equally well as SAMDS in terms of the mean of the solutions found, although SAMDS outperforms HGA-MDS in terms of the attainment rate of reaching the Minimum Dominating Set. Finally, HCSA-MDS outperforms SAMDS, HGA-MDS, and GreedyVoteGr since it is outperformed by HGA-MDS. To conclude, we can state that HCSA-MDS outperforms state-of-the-art approaches in terms of robustness and stability.

	Greedy	GreedyRev	GreedyRan	GreedyVote	GreedyVoteGr	HGA-MDS	SAMDS	HCSA-MDS	
Graph d	Avg(Rea.)	Avg(Rea.)	Avg(Rea.)	Avg(Rea.)	Avg(Rea.)	Avg(Rea.)	Avg(Rea.)	Avg(Rea.)	Time(s)
$N^{400}_{0.1,d}$8	14.2(0)	16.1(4)	30.9(0)	9.2(4)	8(10)	8(10)	8(10)	8(10)	0.09
11	22.1(0)	25.2(0)	33.2(0)	16.8(0)	15.6(5)	11.1(9)	11.1(9)	11(10)	0.07
14	24(0)	25.6(0)	35.2(0)	19(1)	18.6(3)	14.4(6)	14.2(9)	14(10)	0.07
18	27.3(0)	27.3(0)	38.5(0)	21.8(0)	20.4(4)	18.4(6)	18(10)	18(10)	0.11
23	31.3(0)	28.8(0)	41.8(0)	24.9(0)	24.8(0)	24.2(4)	23(10)	23(10)	0.13
$N^{400}_{0.3,d}$3	6.8(0)	9.9(0)	10.9(0)	6.8(4)	6(4)	3(10)	3.8(8)	3.2(8)	0.81
5	9(0)	10.6(0)	11.7(0)	8.7(0)	8.7(0)	5.3(7)	5.2(8)	5(10)	0.15
8	10.9(0)	11.6(0)	13.3(0)	9.3(0)	9.2(0)	8.1(9)	9(8)	8(10)	0.12
11	14(0)	12.6(0)	15.6(0)	11.1(9)	11(10)	11(10)	11(10)	11(10)	0.10
14	16.1(0)	14.2(0)	18.2(0)	14(10)	14(10)	14(10)	14(10)	14(10)	0.10
$N^{400}_{0.5,d}$3	5(0)	6.3(0)	6.4(0)	5(0)	5(0)	3.1(9)	3.1(9)	3(10)	0.48
8	8.8(3)	8.2(8)	9(0)	8.1(9)	8(10)	8(10)	8(10)	8(10)	0.11
11	11.9(3)	11(10)	11.9(4)	11(10)	11(10)	11(10)	11(10)	11(10)	0.06
$N^{800}_{0.1,d}$11	26.3(0)	30.5(0)	40.3(0)	23.7(0)	22.8(1)	11.3(7)	11.3(7)	11(10)	0.45
14	28.1(0)	31.4(0)	41.9(0)	23.5(0)	22.4(2)	14(10)	14(10)	14(10)	0.45
22	34.9(0)	33.1(0)	48.3(0)	27.3(0)	27.3(0)	22.5(5)	22(10)	22(10)	0.60
$N^{800}_{0.3,d}$3	8.7(0)	12.1(0)	12.5(0)	9(0)	8.8(1)	7.9(6)	7.3(6)	3(10)	88.3
5	9.7(0)	12.2(0)	13.3(0)	9.4(0)	9.4(0)	6.8(5)	5(10)	5(10)	1.89
$N^{800}_{0.5,d}$3	6(0)	7.3(0)	7(0)	6(0)	6(0)	4.4(5)	3.4(8)	3.2(9)	3.25
6	7.4(2)	7.8(0)	8.3(0)	6.3(7)	6.2(8)	6.5(8)	6(10)	6(10)	2.41

Table 5: *Numerical results for various approaches on the second benchmark.*

5.4. Runtime analysis and computational complexity

Our method's execution time, relative to instance size, follows a polynomial function $\mathcal{C}(n) = \mathcal{P}(n)$. By conducting polynomial regression on the first benchmark's runtime, we derived an empirical complexity function: $\mathcal{P}(n) = 0.01791n^2 - 0.4981n + 6.206$. This polynomial function adequately represents the

execution time of the best solution found so far. Our approach demonstrates polynomial complexity for the considered instances. The algorithm's theoretical complexity mainly depends on the repair and crossover functions, both with $\mathcal{O}(V^2 + VE)$ complexity, ensuring efficiency and scalability for larger instances (see Figure 5).

6. Conclusion

In this study, we tackled the Minimum Dominating Set problem, a challenging NP-hard problem in graph theory, by proposing a Hybrid Cuckoo Search Algorithm (HCSA-MDS). Our algorithm outperformed state-of-the-art techniques in experimental evaluations on multiple benchmark sets, demonstrating superior performance in best, average, and worst solution comparisons. HCSA-MDS combines the efficient exploration of the cuckoo search algorithm using Lévy flights with intensification schemes and a genetic crossover operator for solution exploitation. As a future perspective, we plan to explore hybridizing with single-solution metaheuristics to further enhance the effectiveness and efficiency of our approach.

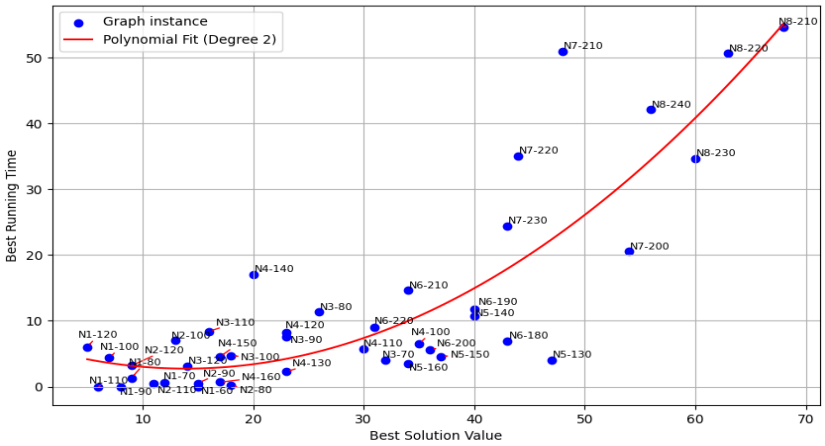


Figure 5: *HCSA-MDS execution time and polynomial fit.*

Acknowledgements

The authors would like to express their gratitude to the dedicated review team of two anonymous reviewers whose insightful comments greatly enhanced the quality of the paper.

References

[1] Abed, S. A. and Rais, H. M. (2017). Hybrid bat algorithm for minimum dominating set problem. Journal of Intelligent & Fuzzy Systems, 33(4), 2329-2339. IOS Press. doi: 10.3233/JIFS-17398

[2] Alkhalifah, Y. and Wainwright, R. L. (2004). A genetic algorithm applied to graph problems involving subsets of vertices. Proceedings of the 2004 Congress on Evolutionary Computation. 04TH8753(1), 303-308. IEEE. doi: 10.1109/CEC.2004.1330871

[3] Jeremy Blum, Min Ding, Andrew Thaeler, Xiuzhen C, Du DZ, and Pardalos P. (2004). Connected dominating set in sensor networks and manets. Handbook of Combinatorial, Springer, US. doi: 10.1007/0-387-23830-1_8

[4] Boumedine, N. and Bouroubi, S. (2022). Protein folding in 3D lattice HP model using a combining cuckoo search with the Hill-Climbing algorithms. Applied Soft Computing, 119, 108564. Elsevier. doi: 10.1016/j.asoc.2022.108564

[5] Brown, C. T., Liebovitch, L. S. and Glendon, R. (2007). Lévy flights in Dobe Ju/'hoansi foraging patterns. Human Ecology, 35, 129-138. Springer. doi: 10.1007/s10745-006-9083-4

[6] Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research, 7, 1-30. JMLR.org. doi: 10.5555/1248547.1248548

[7] Dinh, T. N., Shen, Y., Nguyen, D. T. and Thai, M. T. (2014). On the approximability of positive influence dominating set in social networks. Journal of Combinatorial Optimization, 27(3), 487-503. Springer. doi: 10.1007/s10878-012-9530-7

[8] Garey, M. R. and Johnson, D. S. (1979). Computers and intractability. Freeman, San Francisco. doi: 10.5555/574848

[9] Gandomi, A. H., Yang, X.-S. and Alavi, A. H. (2013). Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Engineering with Computers, 29, 17-35. Springer. doi: 10.1007/s00366-011-0241-y

[10] Haynes, T. W., Hedetniemi, S. and Slater, P. (2013). Fundamentals of domination in graphs. CRC Press. doi: 10.1201/9781482246582

[11] Hedar, A.-R. and Ismail, R. (2010). Hybrid genetic algorithm for minimum dominating set problem. In: Computational Science and Its Applications–ICCSA 2010: International Conference, Fukuoka, Japan, March 23-26, 2010, Proceedings, Part IV, 457-467. Springer. doi: 10.1007/978-3-642-12189-0_40

[12] Hedar, A.-R. and Ismail, R. (2012). Simulated annealing with stochastic local search for minimum dominating set problem. International Journal of Machine Learning and Cybernetics, 3, 97-109. Springer. doi: 10.1007/s13042-011-0043-y

[13] Ho, C. K., Singh, Y. P. and Ewe, H. T. (2006). An enhanced ant colony optimization metaheuristic for the minimum dominating set problem. Applied Artificial Intelligence, 20(10), 881-903. Taylor & Francis. doi: 10.1080/08839510600940132

[14] Yoichi, I. (2011). A faster algorithm for dominating set analyzed by the potential method. IPEC, 2011. doi: 10.1007/978-3-642-28050-4_4

[15] Misra, R. and Mandal, C. (2009). Minimum connected dominating set using a collaborative cover heuristic for ad hoc sensor networks. IEEE Transactions on Parallel and Distributed Systems, 21(3), 292-302. IEEE. doi: 10.1109/TPDS.2009.78

[16] Parekh, A. K. (1991). Analysis of a greedy heuristic for finding small dominating sets in graphs. Information Processing Letters, 39(5), 237-240. Elsevier. doi: 10.1016/0020-0190(91)90021-9

[17] Payne, R. B. and Sorensen, M. D. (2005). The cuckoos. Bird Families of the World, 15. Bird Families of the World. doi: 10.1093/oso/9780198502135.001.0001

[18] Payne, R. and Kirwan, G. (2020) Chestnut-winged Cuckoo (Clamator coromandus). In Del Hoyo, J., Elliott, A., Sargatal, J., Christie, D. and De Juana, E. (Eds.) Birds Of The World, 1st Ed. doi: 10.2173/bow.chwcuc1.01

[19] Sanchis, L. A. (2002). Experimental analysis of heuristic algorithms for the dominating set problem. Algorithmica, 33, 3-18. Springer. doi: 10.1007/s00453-001-0101-z

[20] Pogorelov, K., Schroeder, D. T., Filkuková, P., Brenner, S. and Langguth, J. (2021). Wico text: a labeled dataset of conspiracy theory and 5g-corona misinformation tweets. In: Proceedings of the 2021 Workshop on Open Challenges in Online Social Networks, 21-25. doi: 10.1145/3472720.3483617

[21] Sheskin, D. J. (2003). Handbook of parametric and nonparametric statistical procedures. Chapman and Hall/CRC. doi: 10.1201/9780429186196

[22] Takaguchi, T., Hasegawa, T. and Yoshida, Y. (2014). Suppressing epidemics on networks by exploiting observer nodes. Physical Review E, 90(1), 012807. APS. doi: 10.1103/PhysRevE.90.012807

[23] Umbarkar, A. J. and Sheth, P. D. (2015). Crossover operators in genetic algorithms: a review. ICTACT Journal on Soft Computing, 6(1). doi: 10.5120/ijca2017913370

[24] Walton, S., Hassan, O., Morgan, K. and Brown, M. R. (2011). Modified cuckoo search: a new gradient free optimisation algorithm. Chaos, Solitons & Fractals, 44(9), 710-718. Elsevier. doi: 10.1016/j.chaos.2011.06.004

[25] Wan, P.-J., Alzoubi, K. M. and Frieder, O. (2003). A simple heuristic for minimum connected dominating set in graphs. International Journal of Foundations of Computer Science, 14(02), 323-333. World Scientific. doi: 10.1142/S0129054103001753

[26] Wu, J. and Li, H. (2001). A dominating-set-based routing scheme in ad hoc wireless networks. Telecommunication Systems, 18, 13-36. Springer. doi: 10.1023/A:1016783217662

[27] Xu, Y.-Z. and Zhou, H.-J. (2016). Generalized minimum dominating set and application in automatic text summarization. Journal of Physics: Conference Series, 699(1), 012014. IOP Publishing. doi: 10.1088/1742-6596/699/1/012014

[28] Yang, X.-S. and Deb, S. (2009). Cuckoo search via Lévy flights. In: 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), 210-214. IEEE. doi: 10.1109/NABIC.2009.5393690
[29] Yang, X.-S. and Deb, S. (2014). Cuckoo search: recent advances and applications. Neural Computing and Applications, 24, 169-174. Springer. doi: 10.1007/s00521-013-1367-1
[30] Zhao, D., Xiao, G., Wang, Z., Wang, L. and Xu, L. (2020). Minimum dominating set of multiplex networks: Definition, application, and identification. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 51(12), 7823-7837. IEEE. doi: 10.1109/TSMC.2020.2987163

Exact methods for the longest induced cycle problem

Ahmad Turki Anaqreh[1,*], **Boglárka Gazdag-Tóth**[1] and **Tamás Vinkó**[1]

[1] *University of Szeged, Institute of Informatics, 6720 Szeged, Árpád tér 2, Hungary*
E-mail: ⟨{ahmad, boglarka, tvinko}@inf.u-szeged.hu⟩

Abstract. The longest induced (or chordless) cycle problem is a graph problem classified as NP-complete and involves the task of determining the largest possible subset of vertices within a graph in such a way that the induced subgraph forms a cycle. Within this paper, we present three integer linear programs specifically formulated to yield optimal solutions for this problem. The branch-and-cut algorithm has been used for two models that cannot be directly solved by any MILP solver. To demonstrate the computational efficiency of these methods, we utilize them on a range of real-world graphs as well as random graphs. Additionally, we conduct a comparative analysis against approaches previously proposed in the literature.

Keywords: branch-and-cut algorithm, longest chordless cycle, longest induced cycle, mixed integer linear programming, valid inequalities

Received: March 4, 2024; accepted: July 15, 2024; available online: October 7, 2024

DOI: 10.17535/crorr.2024.0016

1. Introduction

A significant part of combinatorial optimization is closely related to graphs. Within graph theory, the concept of graph cycles has fundamental importance. Identifying a simple cycle or a cycle with a specific structure within a graph forms the basis for numerous graph-theoretical problems that have been under investigation for many years. One such problem is the Eulerian walk, a cyclic path that traverses each edge exactly once, as discussed in [19]. Another example is the Hamiltonian cycle, which traverses every vertex exactly once, as explored in [1].

Kumar et al. [11], introduced a heuristic algorithm for the longest simple cycle problem. The authors utilized both adjacency matrices and adjacency lists, achieving a time complexity for the proposed algorithm proportional to the number of nodes plus the number of edges of the graph. In [2], the authors investigated the longest cycle within a graph with a large minimal degree. For a graph $G = (V, E)$ with a vertex count of $|V| = n$, the parameter $\text{min\_deg}(G)$ denotes the smallest degree among all vertices in G, while $\text{c}(G)$ represents the size of the longest cycle within G. The authors demonstrated that for $n > k \geq 2$, with $\text{min\_deg}(G) \geq n/k$, the lower bound $\text{c}(G) \geq [n/(k-1)]$ holds. Broersma et al. [5] proposed exact algorithms for identifying the longest cycles in claw-free graphs. A claw, in this context, refers to a star graph including three edges. The authors introduced two algorithms for identifying the longest cycle within such graphs containing n vertices: one algorithm operates in $\mathcal{O}(1.6818^n)$ time with exponential space complexity, while the second algorithm functions in $\mathcal{O}(1.8878^n)$ time with polynomial space complexity.

In the work by Lokshtanov [12], the focus lies on the examination of the longest isometric cycle within a graph, which is defined as the longest cycle where the distance between any two

*Corresponding author.

vertices on the cycle remains consistent with their distances in the original graph. The author introduced a polynomial-time algorithm to address this specific problem.

Our primary focus in this paper is dedicated to addressing the challenge of identifying the longest induced (or chordless) cycle problem. For a graph $G = (V, E)$ and a subset $W \subseteq V$, the W-induced graph $G[W]$ comprises all the vertices from set W and the edges from G that connect vertices exclusively within W. The objective of the longest induced cycle problem is to determine the largest possible subset W for which the graph $G[W]$ forms a cycle. While it may seem straightforward to obtain an induced cycle since every isometric cycle is an induced cycle, it has been shown that identifying the longest induced cycle within a graph is an NP-complete problem, as demonstrated by Garey et al. [8].

The longest induced path (P), discussed in [14], represents a sequence of vertices within graph G, where each consecutive pair of vertices is connected by an edge $e \in E$ and there is no edge between non-consecutive vertices within P. In the context of a general graph G, determining the existence of an induced path with a specific length is proven to be NP-complete, as detailed in [8]. Consequently, the longest induced cycle can be considered as a special case of the longest induced path.

Holes in a graph, defined as induced cycles with four or more vertices, play a significant role in various contexts. Perfect graphs, for instance, are characterized by the absence of odd holes or their complements [6]. Moreover, when addressing challenges like finding maximum independent sets in a graph [15], the existence of odd holes leads to the formulation of odd hole inequalities, strengthening approaches for these problems. Similarly, in other problem domains such as set packing and set partitioning [4], these odd hole inequalities serve as crucial components.

Several papers have explored the longest induced cycle problem in graphs with specific structures. In [7], the author investigated the longest induced cycle within the unit circulant graph. To define the unit circulant graph $X_n = Cay(\mathbb{Z}_n; \mathbb{Z}_n^*)$, where n is a positive integer, consider the following. The vertex set of X_n, denoted as $V(n)$, comprises the elements of $\mathbb{Z}_n$, the ring of integers modulo n. The edge set of X_n, represented as $E(n)$, for $x, y \in V(n)$, $(x, y) \in E(n)$ if and only if $x - y \in \mathbb{Z}_n^*$, with $\mathbb{Z}_n^*$ being the set of units within the ring $\mathbb{Z}_n$. The author demonstrates that if the positive integer n has r distinct prime divisors, then X_n contains an induced cycle of length $2^r + 2$. In a separate study by Wojciechowski et al. [20], the authors examine the longest induced cycles within hypercube graphs. If G represents a d-dimensional hypercube, they proved the existence of an induced cycle with a length $\geq (9/64) \cdot 2^d$.

Almost parallel to our work, Pereira et al. [17] dealt with the longest cordless cycle problem, which is equivalent to the longest induced cycle problem. They presented an integer linear program (ILP) formulation along with additional valid inequalities to strengthen and refine the formulation, all of which were incorporated into a branch-and-cut algorithm. They applied a multi-start heuristic method for initial solution generation and then conducted performance evaluations of the algorithm on a range of randomly generated graphs, including those with up to 100 vertices. They could solve the largest problems within 3,011.17 seconds. Our aim is to provide models and methods that can work more efficiently. The models and the best branch-and-cut versions proposed in [17] are discussed in Section 2.5.

Our paper proposes three integer linear programs (ILPs) designed to handle the longest induced cycle problem within general graphs. Some of these models were built based on those used in previous work focused on solving the longest induced path problem, as seen in the studies by Marzo et al. [13] and Bokler et al. [3]. Matsypura et al. [14] introduced three integer programming (IP) formulations and an exact iterative algorithm based on these IP formulations for tackling the longest induced path problem. However, it is important to note that we do not extend these methods, as they were found to be less effective compared to models in [13, 3].

The rest of the paper is organized as follows. Sections 2 and 3 discuss the models and methodologies used to solve the problem together with the best models and methods presented

in [17]. Section 4 reports the numerical results to show the efficiency of our models, also compared to the results in [17]. The conclusion of our work is presented in Section 5.

2. Models

2.1. Notations

Let $G = (V, E)$ be a directed graph with vertex set V and edge set $E \subset V \times V$. An edge $e = (i, j) \in E$ is defined for some $i, j \in V$. The symmetric pair is given as $\bar{e} = (j, i)$. For undirected graphs, we introduce the symmetric edge set $E^* = E \cup \{\bar{e} = (j, i) : e = (i, j) \in E\}$.

We use the notation δ for adjacent edges over vertices and edges as follows. Let us denote the outgoing and incoming edges incident to vertex i with $\delta^+(i) = \{(i, k) \in E^*\}$, and $\delta^-(i) = \{(k, i) \in E^*\}$, respectively. Additionally, $\delta(i) = \delta^+(i) \cup \delta^-(i)$ denote all the edges incident to vertex i.

For an edge $e = (i, j) \in E^*$, outgoing edges are $\delta^+(e) = \delta^+(i) \cup \delta^+(j) \setminus \{e, \bar{e}\}$, and similarly, incoming edges are $\delta^-(e) = \delta^-(i) \cup \delta^-(j) \setminus \{e, \bar{e}\}$. The neighbour edges of e are denoted by $\delta(e) = \delta^+(e) \cup \delta^-(e)$ for all $e \in E^*$. This notation can be extended to any subset of vertices $C \subset V$, where $\delta^+(C) := \{(k, l) \in E^* : k \in C, l \in V \setminus C\}$ and $\delta^-(C) := \{(k, l) \in E^* : l \in C, k \in V \setminus C\}$ denote the outgoing and incoming edges of C, respectively, and $\delta(C) = \delta^+(C) \cup \delta^-(C)$ all edges that connect C with $V \setminus C$.

2.2. Order-based model

The first model to discuss, called *LIC*, is an MILP model using order-based formulation to avoid subtours. The formalism of the model is as follows:

$$\max \sum_{i \in V} y_i \tag{1}$$

subject to

$$x_e + x_{\bar{e}} \leq 1 \qquad \forall\, e \in E \tag{2}$$

$$\sum_{g \in \delta^+(e)} x_g \leq 1 \qquad \forall\, e \in E^* \tag{3}$$

$$y_i = \sum_{g \in \delta^+(i)} x_g \qquad \forall i \in V \tag{4}$$

$$y_i = \sum_{g \in \delta^-(i)} x_g \qquad \forall i \in V \tag{5}$$

$$\sum_{i \in V} w_i = 1 \tag{6}$$

$$w_i \leq y_i \qquad \forall\, i \in V \tag{7}$$

$$u_i - u_j \leq n(1 - x_e) - 1 + n w_i \qquad \forall e \in E^* \tag{8}$$

$$\sum_{i \in V} i w_i \leq j y_j + n(1 - y_j) \qquad \forall\, j \in V \tag{9}$$

$$y_i, u_i \geq 0 \qquad \forall i \in V \tag{10}$$

$$x_e \in \{0, 1\} \qquad \forall e \in E^* \tag{11}$$

$$w_i \in \{0, 1\} \qquad \forall i \in V \tag{12}$$

The variable y_i indicates whether vertex i is part of the longest induced cycle or not. Consequently, the objective in (1) aims to maximize the sum of these variables, which directly corresponds to the length of the cycle. The decision variable x_e is one if the edge e is included in the solution, and zero otherwise.

The constraints can be understood as follows. Given that E^* is symmetric, constraint (2) guarantees that only one of the edges e or $\bar{e}$ can exist in the cycle, preventing the formation of small cycles. Constraint (3) ensures that for any edge $e = (i, j) \in E^*$, only one outgoing edge from either vertex i or vertex j can be part of the cycle. Constraints (4) and (5) ensure that for a given vertex i, only one outgoing edge and one incoming edge can be chosen to be part of the cycle. The variable w_i is introduced to handle the position of the last vertex in the solution, helping in the ordering process. Constraints (6)-(7) ensure that w_i is 1 for exactly one vertex and that w_i can be 1 only for one of the vertices in the solution, respectively. The constraint (8) is a modified Miller-Tucker-Zemlin (MTZ) order-based formulation: if edge $e = (i, j)$ is in the cycle, vertices i and j must be arranged in sequential order according to variables u_i and u_j such that $u_j \geq u_i + 1$, unless the binary variable w_i equals 1. Constraint (9) functions as a symmetry-breaking constraint, as described in [18]. It enforces that the last vertex in the cycle must have the smallest index among all vertices in the cycle.

For a variation of the above introduced *LIC* model consider the following constraint:

$$x_e + x_{\bar{e}} \geq y_i + y_j - 1 \qquad \forall\, e = (i, j) \in E \tag{13}$$

Constraint (13) guarantees that either edge e or $\bar{e}$ must be included in the solution if both endpoints i and j are part of the solution. Conversely, if an edge is not selected for the solution, neither of its endpoints can be included in the solution. By substituting constraint (3) in the original *LIC* model with constraint (13), we create a new model, *LIC*2. This modification leads to improved runtime performance compared to *LIC*, as demonstrated in Section 4.

2.3. Subtour-elimination model

The second model we employ to address the longest induced cycle problem is based on the model presented by Bokler et al. [3], which is referred to ILP_{cut} and was originally designed for identifying the longest induced path. E^* is the symmetric edge set, as defined previously. Let $\mathcal{C}$ represent the set of cycles in G. The model is defined as follows:

$$\max \frac{1}{2} \sum_{e \in E^*} x_e \tag{14}$$

subject to

$$x_e = x_{\bar{e}} \qquad \forall e \in E \tag{15}$$

$$x_e \leq \sum_{g \in \delta^-(i)} x_g \qquad \forall\, e = (i, j) \in E^* \tag{16}$$

$$\sum_{g \in \delta^-(i)} x_g + \sum_{g \in \delta^+(j)} x_g \leq 2 \qquad \forall\, e = (i, j) \in E^* \tag{17}$$

$$\sum_{e \in \delta(i)} x_e \leq \sum_{g \in \delta(C)} x_g \qquad \forall C \in \mathcal{C}, i \in C \tag{18}$$

$$x_e \in \{0, 1\} \qquad \forall e \in E^* \tag{19}$$

The binary decision variable x_e indicates whether edge e is a part of the longest induced cycle, but unlike in the *LIC* model (in Section 2.2), in this case, edge selection is symmetric.

Consequently, the objective is to maximize half of the sum of these variables, as defined in objective function (14). Symmetry of the solution is guaranteed by (15). Constraint (16) enforces that the solution forms a cycle or cycles, while constraint (17) specifies that for any edge e in the solution, precisely two of its adjacent edges must also be in the solution. This ensures the induced property of the solution. Constraint (18) is utilized to eliminate small cycles in the graph.

2.4. Cycle-elimination model

Our third model, called *cec*, is a modified version of the *cec* model introduced in [13] to find the longest induced path. In this model, the symmetry of the edges is not used. The formalism of the model is as follows:

$$\max \sum_{i \in V} y_i \tag{20}$$

subject to

$$\sum_{e \in \delta(i)} x_e = 2y_i \qquad \forall\ i \in V \tag{21}$$

$$x_e \leq y_i \qquad \forall\ i \in V, e \in \delta(i) \tag{22}$$

$$x_e \geq y_i + y_j - 1 \qquad \forall e = (i,j) \in E \tag{23}$$

$$\sum_{i \in C} y_i \leq |C| - 1 \qquad \forall C \in \mathcal{C} \tag{24}$$

$$y_i \in \{0,1\} \qquad \forall i \in V \tag{25}$$

$$x_e \in \{0,1\} \qquad \forall e \in E \tag{26}$$

The binary decision variable y_i maintains its previous interpretation, equal to 1 if vertex i is part of the solution. Additionally, variable x_e is set to 1 if edge e is included in the solution. However, in this context, the symmetric edge is not needed. The objective function (20) seeks to maximize the number of vertices within the induced cycle. Constraint (21) guarantees that each vertex within the solution is connected to precisely two vertices in the cycle. Constraints (22) and (23) are in place to ensure that the cycle is induced. To eliminate solutions composed of small cycles from consideration, constraint (24) is introduced. $\mathcal{C}$ represents a set of the cycles for the given graph. Constraint (24) is added to the model to enforce the solution to consist of a single cycle. This means that multiple small cycles are not deemed valid solutions.

2.5. Cordless-cycle model

The CCP formulation was introduced by Pereira et al. [17] to deal with the problem at hand. The CCP formulation is formally described as follows:

$$\max \sum_{i \in V} y_i \tag{27}$$

subject to

$$\sum_{e\in E} x_e = \sum_{i\in V} y_i \tag{28}$$

$$\sum_{i\in V} y_i \geq 4 \tag{29}$$

$$\sum_{e\in\delta(i)} x_e = 2y_i \qquad \forall\, i \in V \tag{30}$$

$$\sum_{g\in\delta(C)} x_g \geq 2(y_i + y_j - 1) \quad C \subset V, i \in C, j \in V \setminus C \tag{31}$$

$$x_e \leq y_i \qquad \forall\, i \in V, e \in \delta(i) \tag{32}$$

$$x_e \geq y_i + y_j - 1 \qquad \forall\, e = (i,j) \in E \tag{33}$$

$$x_e \in \{0,1\} \qquad \forall e \in E \tag{34}$$

$$y_i \in \{0,1\} \qquad \forall i \in V \tag{35}$$

The formulation includes the usual sets of binary variables: y_i and x_e, indicating whether vertex i and edge e are in the cycle or not. Consequently, the number of selected vertices and edges is equal, as required by (28), and at least four vertices must be selected by (29). Each vertex within the solution is incident to precisely two edges, as guaranteed by (30). Moreover, the subgraph defined by x and y remains connected, as guaranteed by the subtour elimination constraint (31). Furthermore, (32)–(33) ensures that any solution is an induced subgraph of G. More specifically, any edge of G with both its endpoints belonging to the solution must be part of the solution.

The CCP formulation was employed by the authors of [17], along with various valid inequalities. They introduced nine branch-and-cut (BC) algorithms and subsequently chose the top three among them. The first one, labeled as $BC1$ contains constraints (28)-(35), and in addition the following constraint:

$$\sum_{g\in\delta(C)} x_g \geq 2x_e \quad C \subset V, e = (i,j) \in E, i \in C, j \in V \setminus C \tag{36}$$

This algorithm initiates by separating (31), and subsequently, the resulting inequality is enhanced to the more robust form of (36). This specific constraint ensures that if $x_e = 1$, then it is mandatory for $y_i = y_j = 1$ to hold true, due to the presence of inequalities (32)-(33). For the $BC2$ and $BC3$ algorithms, both constraints (37) and (38) were included together with constraints (28)-(36).

$$\sum_{i\in Q} y_i \leq 2 \tag{37}$$

$$\sum_{e\in E(Q)} x_e \geq \sum_{i\in Q} y_i - 1 \tag{38}$$

For a clique $Q \subset V$, $|Q| \geq 3$. Constraint (37) ensures that within a clique Q at most two of its vertices can be part of the induced cycle. On the other hand, constraint (38) guarantees that for a clique Q the number of vertices that can be part of the induced cycle is limited to at most one more than the number of edges that can be included from the clique. Namely, only one of the edges from Q might be included in the solution.

For the $BC2$ algorithm, they implemented a rule that imposes no restrictions on the number of separation rounds. In other words, whenever a violated inequality is detected, it is included

in the cut pool. Conversely, for $BC3$, a fixed number of separation rounds, specifically 30, was established, and inequalities were added to the cut pool if a clique did not share two or more vertices with a clique in a previously accepted inequality. The order of inequalities in the cut pool was determined by descending order of the absolute values of their corresponding linear programming relaxation dual variables. All three algorithms utilized the lower bounds obtained from the multi-start CCP heuristic [17], which is a constructive procedure that takes a predefined edge as input data. The algorithm then seeks to extend a tentative path, P, containing the selected vertices. Vertices are added to P one at a time, accepted if they are adjacent to one of the path's current extremities and not adjacent to any internal vertices. The procedure terminates when the endpoints of P meet, resulting in a chordless cycle of G, or when further expansion of P becomes impossible.

3. Algorithms

Out of the three models we have introduced, only LIC (and $LIC2$) can be directly solved using any MILP solver. Both ILP_{cut} and cec rely on the set of small cycles, which are usually created as part of the solution process, either through an iterative cut generation approach or, more effectively, via branch-and-cut algorithm by employing separation.

Note that subtour elimination inequalities (18) and (24), present in the ILP_{cut} and cec models respectively, exhibit exponential complexity. Consequently, attempting to enumerate all inequalities corresponding to each subtour within the graph and subsequently cutting them becomes impractical. Instead, we have added these inequalities to the ILP_{cut} and cec models as soon as facing them. Hence, the cut generation approach is employed as follows: the method is initiated with a model relaxing all subtour elimination inequalities, and if subtours arise in integer solutions, violated inequalities are added, and this process is repeated until the optimal solution is reached. For that, callback functionality from Gurobi [9] was employed, which can be used to add these inequalities iteratively.

We employed the Depth-First Search (DFS) algorithm on the induced subgraph of the integer solution to identify cycles, subsequently introducing a new inequality for each subtour discovered.The entire procedure, which combines the models and cut generations, is shown in Algorithms 1 and 2.

3.1. Initialization

In the initialization phase of the procedure, the ILP_{cut} and cec models are created, encompassing the creation of their variables, constraints, and objective functions.

3.2. Cut generation

To tackle the ILP_{cut} and cec models, we combine the cut generation mechanism with the Branch-and-bound method, as explained earlier in this section. Consequently, each model was addressed using two distinct methodologies, as outlined below.

3.2.1. Soft approach

The first approach involves cut generation as outlined in Algorithm 1. In each iteration, a subproblem from the Branch-and-Bound tree is solved. In line 4 the algorithm checks if the solution of the subproblem is an integer solution. Based on this, the DFS algorithm is employed to detect any subtours within the solution, as shown in line 5. If a subtour exists, and its length is less than or equal to the value of the variable `longest_induced_cycle`, a cut is appended for that cycle. If not, the value of the variable is updated to reflect the length of the cycle, and

there is no need to introduce a cut because the cycle could potentially be the optimal solution. These details are clarified in lines 6 through 9. The cut generation terminates when there are no further subtours present in the solution, indicating the completion of the procedure.

Algorithm 1 Cut_Generation1

```
1: Input: model Initialization()
2: longest_induced_cycle ← 0
3: function CUT_GENERATION1()
4:     if model.status == feasible_integer then          ▷ model has integer solution
5:         C ← DFS(feasible_integer)                     ▷ find subtour in the solution
6:         if length(C) ≤ longest_induced_cycle then
7:             model.addConstr(18, 24)                   ▷ add cut (18) or (24) to the model
8:         else
9:             longest_induced_cycle ← length(C)         ▷ update variable value
10:        end if
11:    end if
12: end function
13: model.optimize(Cut_Generation1())                    ▷ solve the model using cut generation
14: print(longest_induced_cycle)
```

3.2.2. Tough approach

The second cut generation-based approach is detailed in Algorithm 2. In each iteration, a subproblem is solved, and if an integer solution is obtained, the algorithm verifies the presence of any subtours using the DFS algorithm, as described in lines 4 through 5. If any cycles are detected, a cut is integrated into the model (line 6), and the length of the cycle is updated if it exceeds the value of the variable `longest_induced_cycle` (line 8). Although we may cut the optimal induced cycle, its length (and possibly the cycle itself) is recorded. It is important to note that in order to further improve the procedure, a constraint is added to the model in line 10. This constraint ensures that the objective value must be greater than the length of the largest induced cycle discovered so far. By using this cut generation, the branch-and-cut indicates the infeasibility of the problem, yet the longest induced cycle length recorded in the variable `longest_induced_cycle`.

Algorithm 2 Cut_Generation2

```
1: Input: model Initialization()
2: longest_induced_cycle ← 0
3: function CUT_GENERATION2()
4:     if model.status == feasible_integer then          ▷ model has integer solution
5:         C ← DFS(feasible_integer)                     ▷ find subtour in the solution
6:         model.addConstr(18, 24)                       ▷ add cut (18) or (24) to the model
7:         if length(C) > longest_induced_cycle then
8:             longest_induced_cycle ← length(C)
9:         end if
10:        model.addConstr(model.ObjVal >= longest_induced_cycle + 1)
11:    end if
12: end function
13: model.optimize(Cut_Generation2())                    ▷ solve the model using cut generation
14: print(longest_induced_cycle)
```

3.3. Longest Isometric Cycle

Lokshtanov's algorithm, as described in [12], aims to identify the longest isometric cycle within a graph. In accordance with the definition of an isometric cycle, as discussed in Section 1, if a given graph G contains an isometric cycle with a length of ℓ, then there must also exist an induced cycle within the graph with a length of m where $m \geq \ell$. Consequently, the longest isometric cycle serves as a benchmark for the longest induced cycle. The algorithm's objective is to verify the existence of an isometric cycle with a length of k in a given graph $G = (V, E)$. If such a cycle exists, the graph G can be employed to construct a new graph G_k with vertices as vertex-pairs of G. Namely, $V(G_k) = \{(u, v) \in V : d(u, v) = \lfloor k/2 \rfloor\}$, where $d(u, v)$ is the length of the shortest path between u and v, and its edge set given by $E(G_k) = \{((u, v), (w, x)) : (u, w) \in E(G) \wedge (v, x) \in E(G)\}$.

The method is outlined in Algorithm 3. For a given value of k, the algorithm computes the graph G_k and examines whether there exists a pair of vertices (u, v) and (v, x) within $V(G_k)$ such that (v, x) belongs to the set $M_k(u, v) := \{(u, x) : (u, x) \in V(G_k) \wedge (v, x) \in E(G)\}$ and $d_{G_k}[(u, v), (v, x)] = \lfloor k/2 \rfloor$. If such a pair is found, it indicates the presence of an isometric cycle with a length of k.

Algorithm 3 Longest Isometric Cycle

```
1: LISC ← 0
2: for all l ∈ V, i ∈ V, j ∈ V do                ▷ distance calculation by Floyd algorithm
3:     d(i, j) ← min{d(i, j), d(i, l) + d(j, l)}
4: end for
5: if G is a tree then                           ▷ no cycles in tree graph
6:     return LISC
7: end if
8: for k = 3 → n do
9:     V_k ← ∅                                   ▷ vertices of G_k
10:    for all u, v ∈ V do
11:        if d(u, v) = ⌊k/2⌋ then
12:            V_k ← V_k ∪ {(u, v)}
13:        end if
14:    end for
15:    E_k ← ∅                                   ▷ edges of G_k
16:    for all (u, v), (w, x) ∈ V_k do
17:        if (u, w) ∈ E ∧ (v, x) ∈ E then
18:            E_k ← E_k ∪ {((u, v), (w, x))}
19:        end if
20:    end for
21:    G_k ← (V_k, E_k)
22:    for all (u, v, x) ∈ V do
23:        if (u, v) ∈ V_k ∧ (v, x) ∈ M_k(v, u) ∧ d_{G_k}[(u, v), (v, x)] = ⌊k/2⌋ then
24:            LISC ← k
25:        end if
26:    end for
27: end for
28: print(LISC)
```

4. Numerical experiments

To demonstrate and evaluate the effectiveness of the proposed methods, we present numerical results for three models: the *LIC* model, the ILP_{cut} model, and the *cec* model. Furthermore, we conducted a comparison between our best results and results from [17] on randomly generated graphs to highlight the efficiency of our approach in comparison to existing methods.

4.1. Computational environment and datasets

The algorithms detailed in Section 3 were implemented in Julia 1.7.0, utilizing the JuMP package version 0.22.1. We employed Gurobi 9.5.0 as the solver for all experiments. Each run was constrained to a one-hour time limit and a single thread. For the longest isometric cycle algorithm, we implemented it using Python 3.8 with a 24-hour time limit. These computations were performed on a computer with an Intel Core i7-4600U CPU, 8GB of RAM, and running the Windows 10 operating system.

To verify the efficacy of our methods, we employed two sets of network datasets. The first is the RWC set, comprising 19 real-world networks that encompass communication and social networks within companies, networks of book characters, as well as transportation, biological, and engineering networks, as described in [14]. Additionally, we utilized the Movie Galaxy (MG) set, consisting of 773 graphs that represent social networks among movie characters, as detailed in [10]. For further information about these instances, the reader is referred to the following link: http://tcs.uos.de/research/lip.

To perform a comparison with the results presented in [17], we conducted experiments on random graphs with varying values of n ranging from 50 to 100, considering both 10% and 30% density, as in [17]. For every case, 10 graphs were generated. Every run was restricted to a maximum duration of one hour, with no restrictions on the number of threads, and with an initial solution set to 4, as described in [17]. Regarding the hardware comparison, we utilized the information available in [16] to collect the details of the CPU utilized in all experiments, as outlined in Table 1. It is evident that the computer used in [17] is more powerful than ours. To ensure a fair comparison, we normalized the execution times in all cases. The ratio between the single-thread ratings gives a good approximation of the relative speed. Therefore, we calculated this ratio in the last row of Table 1. The run time was then modified by multiplying it by the obtained ratio.

Benchmarks	Intel Core i7-4600U	Intel Xeon W-3223 [17]
Clock Speed (GHz)	2.1	3.5
Turbo Speed (GHz)	Up to 3.3	Up to 4.0 GHz
Number of Physical Cores	2 (Threads: 4)	8 (Threads: 16)
Single Thread Rating	1641	2480
Ratio	0.66	1

Table 1: *CPU performance comparison between the CPU used in this paper and in [17].*

4.2. Computational results

Table 2 presents the computational experiments conducted on the RWC instances. The second column displays the optimal solutions for each instance (opt). In the third column, we find the length of the longest isometric cycle (*LISC*), if possible. The fourth and fifth columns respectively indicate the number of vertices (N) and edges (M) of the corresponding graph.

Columns six through eleven show the time in seconds required to identify the optimal solution using the various methods employed in this study. Specifically, $ILP_{cut}2$ and $cec2$ refer to the methods outlined in Algorithm 2. For all these methods, we also initiated the search using the *LISC* value, incorporating the constraint `ObjVal` $\geq$ `LISC`. These methods are indicated in every second row corresponding to each graph. Instances that resulted in timeouts are denoted by the symbol 🕓.

graph	opt	LISC	N	M	LIC	$LIC2$	ILP_{cut}	$ILP_{cut}2$	cec	$cec2$
high-tech	10	5	33	91	1.22	0.63	0.95	0.33	0.38	0.14
					0.99	0.42	1.15	1.57	1.02	0.77
karate	6	5	34	78	0.63	0.58	0.21	0.24	0.20	0.19
					0.66	0.53	0.23	0.29	0.24	0.17
mexican	13	7	35	117	0.92	0.82	0.66	0.88	0.24	0.20
					0.83	0.78	0.69	0.74	0.20	0.20
sawmill	6	5	36	62	0.54	0.43	**0.18**	0.37	**0.15**	0.10
					0.33	0.30	**0.36**	0.16	**0.13**	0.11
tailorS1	12	7	39	158	2.93	1.11	1.37	1.45	0.34	0.33
					1.38	0.89	1.76	1.76	0.37	0.44
chesapeake	15	5	39	170	1.01	0.69	0.56	0.81	0.24	0.22
					1.05	0.72	0.97	0.87	0.28	0.31
tailorS2	12	5	39	223	3.11	2.05	3.49	4.46	0.74	0.65
					3.25	3.09	3.74	4.62	0.83	0.75
attiro	28	9	59	128	0.93	1.24	0.55	0.53	0.18	0.24
					0.67	0.71	0.52	0.68	0.28	0.31
krebs	8	7	62	153	10.91	7.23	**1.19**	0.94	**0.94**	0.48
					10.39	5.56	**0.86**	1.02	**0.57**	0.38
dolphins	20	7	62	159	14.75	23.70	1.81	2.35	1.70	1.02
					10.66	13.83	2.80	2.98	0.74	1.50
prison	28	9	67	142	5.90	10.22	**0.83**	1.56	**0.62**	0.61
					6.93	10.23	**4.81**	1.03	**0.66**	0.48
huck	5	5	69	297	519.95	299.12	**19.53**	17.79	**4.31**	4.51
					447.72	493.74	**18.22**	19.71	**3.34**	3.31
sanjuansur	35	11	75	144	6.16	5.85	0.68	0.71	0.37	0.49
					7.70	3.61	0.82	1.36	0.44	0.38
jean	7	5	77	254	276.98	147.77	15.93	14.57	2.45	2.41
					150	147.85	13.97	14.80	2.32	2.41
david	15	8	87	406	544.99	308.06	46.23	54.23	5.22	4.36
					219.14	323.96	45.24	38.33	3.31	3.47
ieeebus	32	13	118	179	2.52	5.14	0.76	0.94	0.43	0.62
					7.82	5.82	0.79	1.35	0.33	0.30
sfi	3	3	118	200	6.98	6.51	**0.74**	0.90	**0.3**	0.31
					6.40	2.80	**0.84**	1.32	**0.34**	0.31
anna	15	🕓	138	493	90.75	52.11	10.60	23.65	1.37	1.71
					-	-	-	-	-	-
494bus	116	🕓	494	586	108.48	126.77	27.13	33.09	2.73	2.10
					-	-	-	-	-	-
average					84.19	52.63	7.02	8.41	1.21	1.09
					51.52	59.70	5.75	5.45	0.9	0.92

Table 2: *Running times on RWC instances, time is given in seconds.*

The various methods exhibit diverse performance characteristics in terms of execution time and the number of instances solved optimally. Key observations from Table 2 are as follows:

- *cec*2 outperforms *cec* in 13 cases, ILP_{cut}, $ILP_{cut}2$, *LIC* and *LIC*2 in all the cases.
- $ILP_{cut}2$ was faster than *LIC*2 for 15 cases and *LIC* for all instances.
- *LIC*2 outperforms *LIC* in 14 cases.
- For some instances in ILP_{cut} and *cec*, the graphs and results are indicated by boldface and underlined in Table 2. This is to emphasize that these graphs contain multiple longest induced cycles of the same length, and the procedures described in Algorithm 1 cut the cycle if its length is less than or equal to the longest induced cycle found so far. Thus, for these graphs, all the longest cycles are found by the method.
- Using *LISC* as an initial solution does not contribute significantly to improving the execution time in the majority of cases.
- The results emphasize the correlation between graph density and execution time. Graph density is defined as the ratio of the edges present in a graph to the maximum number of edges it can hold. This relationship is particularly evident for dense graphs like huck, jean, and david, especially in the case of *LIC* and *LIC*2 models. However, it is not the case for *cec* and *cec*2 models as their running times show less sensitivity to the graph's density.

nr. of edges	nr. of instances	*LIC*	*LIC*2	ILP_{cut}	$ILP_{cut}2$	*cec*	*cec*2
1–49	107	0.14	0.14	0.12	0.18	0.09	0.08
50–74	135	0.37	0.29	0.2	0.3	0.17	0.12
75–99	151	0.7	0.58	0.32	0.39	0.22	0.17
100–124	121	1.74	1.32	0.51	0.65	0.29	0.24
125–149	90	4.5	3.67	0.82	0.99	0.37	0.34
150–199	89	10.45	7.48	1.49	1.7	0.56	0.49
200–629	80	169.38	110.49	13.28	16.39	2.41	1.9
average		157.23	126.35	44.73	57.28	26.84	22.01

Table 3: *Running times on MG instances, time is given in seconds.*

The results for the MG instances, organized into groups based on the number of edges, are presented in Table 3. Unlike *LIC* and *LIC*2, where the running times increase proportionally with the instance size, the results indicate that *cec* and *cec*2 are more reliable, with running times showing less sensitivity to the graph's size.

The results for the random graphs are presented in Table 4, where we compare *cec*2 against the top three algorithms introduced in [17]. The runtime represents the average duration on the ten graphs in each case. Notably, *cec*2 outperforms these algorithms in all cases, even before normalizing the execution times with the ratio listed in Table 1. Moreover, *cec*2 successfully solved the instance with 100 vertices and 30% density, a scenario where none of the algorithms in [17] succeeded.

	Randomly generated graphs: 10% density			
n	*cec*2	$BC1$	$BC2$	$BC3$
50	0.32 (0.49)	0.33	0.3	0.39
60	0.79 (1.2)	1.19	1.2	1.34
70	4.17 (6.32)	5.57	4.93	5.58
80	20.5 (31.1)	37.15	26.9	27.34
90	93.93 (142.32)	160.1	155.82	168.02
100	518.75 (785.99)	1321.41	1129.47	1094.8
average	106.41	254.29	219.77	216.25
	Randomly generated graphs: 30% density			
n	*cec*2	$BC1$	$BC2$	$BC3$
50	4.15 (6.28)	8.21	9.6	8.78
60	26.14 (39.6)	39.82	46.18	51.9
70	90.11 (136.52)	234.45	206.36	283.28
80	203.81 (308.8)	935.78	676.52	1139.55
90	544.88 (825.58)	2072.16	1874.91	3011.17
100	1810.49 (2743.17)			
average	446.6	658.08	562.71	898.94

Table 4: *Running times on random instances for* cec*2,* BC*1,* BC*2, and* BC*3, time is given in seconds. Values in brackets show the original execution time of cec2.*

5. Conclusion

Considering that the longest induced cycle problem is NP-hard, it is essential to find an efficient approach that can yield optimal solutions within a reasonable time. In this regard, we introduced three integer linear programs, some of which are extensions of models originally formulated for solving the longest induced path problem. These newly proposed programs showed differing execution times and success rates in terms of achieving optimal solutions, outperforming the models presented in the literature. We have found that the *cec*2 formulation with tough cut generation yields the most efficient method. For future work, heuristic or metaheuristic approaches can be used as initial solutions for the MILP, potentially increasing the efficiency of solving the problem. Designing more sophisticated MILP formulations with constraints that can more effectively prune the search space is another possibility. Additionally, exploring exact methods like branch-and-price algorithms, which combine branch-and-bound with column generation, could be effective in handling large and complex instances.

Acknowledgements

The research leading to these results has received funding from the national project TKP2021-NVA-09. Project no. TKP2021-NVA-09 has been implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the TKP2021-NVA funding scheme. The work was also supported by the grant SNN-135643 of the National Research, Development and Innovation Office, Hungary.

References

[1] Akiyama, T. and Nishizeki, T. and Saito, N. (1980). NP-completeness of the Hamiltonian cycle problem for bipartite graphs. Journal of Information Processing, 3(2), 73-76.

[2] Alon, N. (1986). The longest cycle of a graph with a large minimal degree. Journal of Graph Theory, 10(1), 123-127. doi: 10.1002/jgt.3190100115

[3] Bökler F., Chimani M., Wagner M. H. and Wiedera T. (2020). An experimental study of ILP formulations for the longest induced path problem. International Symposium on Combinatorial Optimization, 89-101. doi: 10.1007/978-3-030-53262-8_8

[4] Borndörfer, R. (1998). Aspects of set packing, partitioning, and covering. PhD thesis, TU Berlin, Berlin.

[5] Broersma, H., Fomin, F. V., van't Hof, P. and Paulusma, D. (2013). Exact algorithms for finding longest cycles in claw-free graphs. Algorithmica, 65(1), 129-145. doi: 10.1007/s00453-011-9576-4

[6] Chen, Y. and Flum, J. (2007). On parameterized path and chordless path problems. Twenty-Second Annual IEEE Conference on Computational Complexity (CCC'07), 250–263. doi: 10.1109/CCC.2007.21

[7] Fuchs, E. D. (2005). Longest induced cycles in circulant graphs. The Electronic Journal of Combinatorics, Article Number R52. doi: 10.37236/1949

[8] Garey, M. R. and Johnson, D. S. (1990). Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co.

[9] Gurobi Optimization Inc. Gurobi Optimizer Reference Manual (2020).

[10] Kaminski, J., Schober, M., Albaladejo, R., Zastupailo, O. and Hidalgo, C. (2018). Moviegalaxies-social networks in movies. Retrieved from: cris.maastrichtuniversity.nl

[11] Kumar, P. and Gupta, N. (2014). A heuristic algorithm for longest simple cycle problem. Proceedings of the International Conference on Wireless Networks (ICWN), 202-208.

[12] Lokshtanov, D. (2009). Finding the longest isometric cycle in a graph. Discrete Applied Mathematics, 157(12), 2670-2674. doi: 10.1016/j.dam.2008.08.008

[13] Marzo, R., Melo, R. A., Ribeiro, C. C. and Santos, M. C. (2022). New formulations and branch-and-cut procedures for the longest induced path problem. Computers & Operations Research, 139, 105627. doi:10.1016/j.cor.2021.105627

[14] Matsypura, D., Veremyev, A., Prokopyev, O. A. and Pasiliao, E. L. (2019). On exact solution approaches for the longest induced path problem. European Journal of Operational Research, 278(2), 546-562. doi: 10.1016/j.ejor.2019.04.011

[15] Nemhauser, G. L. and Sigismondi, G. (1992). A strong cutting plane/branch-and-bound algorithm for node packing. Journal of the Operational Research Society, 43(5), 443-457. doi: 10.1057/jors.1992.71

[16] PassMark Software - CPU Benchmarks (2023). Retrieved from: cpubenchmark.net/compare

[17] Pereira, D. L., Lucena, A., Salles da Cunha, A. and Simonetti, L. (2022). Exact solution algorithms for the chordless cycle problem. INFORMS Journal on Computing, 34(4),1970-1986. doi: 10.1287/ijoc.2022.1164

[18] Walsh, T. (2012). Symmetry breaking constraints: Recent results. Proceedings of the AAAI Conference on Artificial Intelligence, 26(1), 2192-2198. doi: 10.1609/aaai.v26i1.8437

[19] West, D. B. (2001). Introduction to Graph Theory. Prentice Hall.

[20] Wojciechowski, J. M. (1990). Long induced cycles in the hypercube and colourings of graphs. University of Cambridge.

Croatian Operational Research Review

Information for authors

Journal homepage: http://www.hdoi.hr/crorr-journal

Aims and Scope: The aim of the Croatian Operational Research Review journal is to provide high quality scientific papers covering the theory and application of operational research and related areas, mainly quantitative methods and machine learning. The scope of the journal is focused, but not limited to the following areas: combinatorial and discrete optimization, integer programming, linear and nonlinear programming, multiobjective and multicriteria programming, statistics and econometrics, macroeconomics, economic theory, games control theory, stochastic models and optimization, banking, finance, insurance, simulations, information and decision support systems, data envelopment analysis, neural networks and fuzzy systems, and practical OR and application.

Papers blindly reviewed and accepted by two independent reviewers are published in this journal.

Author Guidelines: There is no submission fee for publishing papers in this journal. All manuscripts should be submitted via our online manuscript submission system at the journal homepage. A user ID and password can be obtained on the first visit. The initial version of the manuscript should be written using LaTex program or Word program, and submitted as a single PDF file. After the reviewing process, if the paper is accepted for publication in this journal, the corresponding author will be asked to submit proofreading confirmation along with signed copyright form. CRORR uses the Turnitin Similarity Check plagiarism software to detect instances of overlapping and similar text in submitted manuscripts. Any copying of text, figures, data or results of other authors without giving credit is defined as plagiarism and is a breach of professional ethics. Such papers will be rejected and other penalties may be accessed.

Electronic Presentations: The journal is also presented electronically on the journal homepage. All issues are also available at the Hrcak database – Portal of science journals od the Republic of Croatia, with full-text search of individual, at

http://hrcak.srce.hr/crorr

Offprints: The corresponding author of the published paper will receive a single print copy of the journal. All published papers are freely available online.

Publication Ethics and Malpractice Statement

Croatian Operational Research Review journal is committed to ensuring ethics in publication and quality of articles. Conformance to standards of ethical behaviour is therefore expected of all parties involved: Authors, Editors, Reviewers, and the Publisher. Our ethic statements are based on COPE's Best Practice Guidelines for Journals Editors.

Publication decisions. The editor of the journal is responsible for deciding which of the articles submitted to the journal should be published. The editor may be guided by the policies of the journal's editorial board and constrained by such legal requirements as shall then be in force regarding libel, copyright infringement and plagiarism. The editor may confer with other editors or reviewers in making this decision.

Fair play. An editor will at any time evaluate manuscripts for their intellectual content without regard to race, gender, sexual orientation, religious belief, ethnic origin, citizenship, or political philosophy of the authors.

Confidentiality. The editor and any editorial staff must not disclose any information about a submitted manuscript to anyone other than the corresponding author, reviewers, potential reviewers, other editorial advisers, and the publisher, as appropriate.

Disclosure and conflicts of interest. Unpublished materials disclosed in submitted manuscript must be used in an editor's own research without the express written consent of the author.

Duties of Reviewers

Contribution to Editorial Decisions. Peer review assists the editor in making editorial decisions and through the editorial communications with the author may also assist the author in improving the paper.

Promptness. Any selected referee who feels unqualified to review the research reported in a manuscript or knows that its prompt review will be impossible should notify the editor and excuse himself from the review process.

Confidentiality. Any manuscripts received for review must be treated as confidential documents. They must not be shown to or discussed with others except at authorized by the editor.

Standards of Objectivity. Reviews should be conducted objectively. Personal criticism of the author is inappropriate. Referees should express their views clearly with supporting arguments.

Acknowledgement of Sources. Reviewers should identify relevant published work that has not been cited by the authors. Any statement that an observation, derivation, or argument had been previously reported should be accompanied by the relevant citation. A reviewer should also call to the editor's attention any substantial similarity or overlap between the manuscript under consideration and any other published paper of which they have personal knowledge.

Disclosure and Conflict of Interest. Privileged information or ideas obtained through peer review must be kept confidential and not used for personal advantage. Reviewers should not consider manuscripts in which they have conflicts of interest resulting from competitive, collaborative, or other relationships or connections with any of the authors, companies, or institutions connected to the papers.

Duties of Authors

Reporting Standards. Authors' manuscripts of original research should present an accurate of the work performed as well as an objective discussion of its significance. Underlying data should be represented accurately in the paper. A paper should contain sufficient detail and references to permit others to replicate the work. Fraudulent or knowingly inaccurate statements constitute unethical behaviour are unacceptable.

Data Access and Retention. When submitting their article for publication in this journal, the authors confirm the statement that all data in article are real and authentic. Authors should be prepared to provide the raw data in connection with a paper for editorial review, and should be prepared to provide public access to such data (consistent with the ALPSP-STM Statement on Data and Databases), if practicable, and should in any event be prepared to retain such data for a reasonable time after publication. In addition, all authors are obliged to provide retractions or corrections of mistakes if any.

Originality and Plagiarism. The authors should ensure that they have written entirely original works, and if the authors have used the work and/or words of others that has been appropriately cited or quoted. Multiple, Redundant or Concurrent Publication An author should not in general publish manuscripts describing essentially the same research in more than one journal or primary publication. Submitting the same manuscript to more than one journal concurrently constitutes unethical publishing behaviour and is unacceptable.

Acknowledgement of Sources. Proper acknowledgement of the work of others must always be given. Authors should cite publications that have been influential in determining the nature of the reported work.

Authorship of the Paper. Authorship should be limited to those who have made a significant contribution to the concept, design, execution, or interpretation of the reported study. All those who have made significant contributions should be listed as co-authors. Where there are others who have participated in certain substantive aspects of the research project, they should be acknowledged or listed as contributors. The corresponding author should ensure that all appropriate co-authors and no inappropriate co-authors are included on the paper, and that all co-authors have seen and approved the final version of the paper and have agreed to its submission for publication.

Disclosure and Conflicts of Interest. All authors should disclose in their manuscript any financial or other substantive conflict of interest that might be construed to influence the results or interpretation of their manuscript. All sources of financial support for the project should disclosed. When an author discovers a significant error or inaccuracy in his/her own published work, it is the author's obligation to promptly notify the journal editor or publisher and cooperate with editor to retract or correct the paper.

Operational Research

Operational Research (OR) is an interdisciplinary scientific branch of applied mathematics that uses methods such as mathematical modelling, statistics, and algorithms to achieve optimal or near optimal solutions to complex problems. It is also known as Operations Research, and is related to Management Science, Industrial Engineering, Decision Making, Statistics, Simulations, Business Analytics, and other areas.

Operational Research is typically concerned with optimizing some objective function, or finding non-optimal but satisfactory solutions. It generally helps management to achieve its goals using scientific methods. Some of the primary tools used by operations researchers are optimization, statistics, probability theory, queuing theory, game theory, graph theory, decision analysis, and simulation.

Areas/Fields in which Operational Research may be applied can be summarized as follows:

- Assignment problem,
- Business analytics,
- Decision analytics,
- Econometrics,
- Linear and nonlinear programming,
- Inventory theory,
- Optimal maintenance,
- Scheduling,
- Stochastic process, and
- System analysis.

Operational Research has improved processes in business and all around us. It is used in everyday situations, such as design of waiting lines, supply chain optimization, scheduling of buses or airlines, telecommunications, or global resources planning decisions.

Operational Research is an important area of research because it enables the best use of available resources. New methods and models in operational research are to be developed continuously in order to provide adequate solutions to process enormous information growth resulted from rapid technology development in today's new economy. By improving processes it enables high-quality products and services, better customer satisfaction, and better decision making. There is no doubt that operational research contributes to the quality of life and economic prosperity on microeconomic, macroeconomic, and global level.

Croatian Operational Research Society

Croatian Operational Research Society (CRORS) was established in 1992. as the only scientific association in Croatia specialized in operational research. The Society has more than 150 members and its main mission is to promote operational research in Croatia and worldwide for the benefit of science and society. This mission is realized through several goals:

- to encourage collaboration of scientists and researchers in the area of operational research in Croatia and worldwide through seminars, conferences, lectures and similar ways of collaborations,
- to organize the International Conference on Operational Research (KOI),
- to publish a scientific journal Croatian Operational Research Review (CRORR).

Since 1994, CRORS is a member of The International Federation of Operational Research Societies (IFORS), an umbrella organization comprising the national Operations Research societies of over forty-five countries from four geographical regions: Asia, Europe, North America and South America. CRORS is also a member of the Association of European Operational Research Societies (EURO) and actively participates in international promotion of operational research.

If you are a researcher, academic teacher, student or practitioner interested in developing and applying operations research methods, you are welcome to become a member of the CRORS and join our OR community.

Former Presidents:	Prof. Luka Neralić, PhD, University of Zagreb (1992-1996) Prof. Tihomir Hunjak, PhD, University of Zagreb (1996-2000) Prof. Kristina Šorić, PhD, University of Zagreb (2000-2004) Prof. Valter Boljunčić, PhD, University of Pula (2004-2008) Prof. Zoran Babić, PhD, University of Split (2008-2012) Prof. Marijana Zekić-Sušac, PhD, University of Osijek (2012-2016) Prof. Zrinka Lukač, PhD, University of Zagreb (2016-2020) Prof. Mario Jadrić, PhD, University of Split (2020-2022)
President:	assistant professor Tea Šestanović, PhD, University of Split (since 2022)
Vice President:	Prof. Snježana Pivac, PhD, University of Split
Secretary:	Tea Kalinić Milićević, mag. math., University of Split
Treasurer:	assistant professor Marija Vuković, PhD, University of Split

If you want to become a member of the CRORS, please fill in the application form available at

http://www.hdoi.hr