# An Opposition-based Genetic Algorithm for Multi-path Routing Problem with Risk

Somnath Maji[1], Samir Maity[2], Debasis Giri[3], Manoranjan Maiti[4]

Dept. of Computer Science and Engineering, MAKAUT, Nadia, W.B., India[1]

Data Science, University of Kalyani, Nadia, W.B., India[2]

Department of Information Technology, MAKAUT, Nadia, W.B., India[3]

Former professor, Dept. of Mathematics, Vidyasagar University, W.B., India[4]

Corresponding author: Somnath Maji, Email: somnathmajivucs@gmail.com

Here, we propose an Opposition-based Genetic Algorithm (OBGA) to design and solve a Multi-path Routing Problem with Risk (MPRPwR). We consider a person purchasing some precious products (jewellery, electronic gadgets, etc.) from some different shops/wholesalers at different places with predetermined demands and having some risks of theft along the routes of routing. Here, two types of risks-one depending on distance and amount of valuable materials and other pathwise are considered. Thus the problem is to find a round trip starting from and ending to the depot after visiting all the shops and collecting the precious products at a minimum system cost under a risk constraint. Here we introduce different alternate paths for travel between the shops. To solve it, OBGA with probabilistic selection, comparison crossover and generation dependent opposition-based mutation is developed and tested against some standard test functions. The effectiveness of our model (MPRPwR) solved by the proposed algorithm (OBGA) is illustrated. A standard Genetic Algorithm (SGA) is used for comparison with OBGA. As particular cases, the model has been solved with single path and path-dependent risks.

**Keywords**: Opposition-based learning, Genetic Algorithm (GA), Cash in transit, Risk of theft, Fixed charge.

2021. In Rahul Srivastava & Aditya Kumar Singh Pundir (eds.), *New Frontiers in Communication and Intelligent Systems*, 727–746. Computing & Intelligent Systems, SCRS, India. https://doi.org/10.52458/978-81-95502-00-4-73

# 1 Introduction

## 1.1 Motivation

In India, after Foreign Direct Investment (FDI), nowadays, business centres (shopping mall, retail shop, etc.) are being established in remotely located urban areas [6]. In developing countries, robbery is a very common phenomenon during journey/transportation. Purchasing of high valued products such as ornaments, jeweleries, etc. from different remotely located shops suppliers is challenged due to robbery, etc. [13] and involves some risks. Normally, people make this type of purchases under some risk. Through out the world, due to the development of infrastructure, nowadays alternate paths/connections among the different places are available. In India, National Highways (NHs) and prime minister Yojana Roads have connected different cities and villages. To maintain these roads, some taxes (called fixed charge, say) are collected at some places on the roads. These facts prompted us to consider the following fixed charge cash-in-transit problem.

A person starts with a car from his town and purchases some valuable ornaments as per predetermined demands from some fixed places (shops) and comes back to his place. During the journey, there are several connecting roads between different destinations (places), and risks of robberies are involved on some routes. The person plans to operate the tour under a specified risk and the problem is to find the optimum path under the specified risk so that total system cost (travel cost+loading cost+fixed charge).

This is a fixed charge routing problem, which is the generalization of TSP type problems. This may also be considered as a pick-up/delivery/cash-in-transit problem with risk [3], [11]. Basically, it is a routing problem of NP-hard nature.

As the proposed Multi-path Routing Problem with Risk (MPRPwR) is of $NP$-hard nature, to solve it, we develop an opposition-based genetic algorithm (OBGA) with a probabilistic selection, comparison crossover and opposition based (OB) mutation.

Opposition-based learning (OBL) was first investigated by [14]. The probability using OBL to get optimal solution, is more than the other methods [15]. In the concept of OBL, first we evaluate the solutions and their opposite solutions and from those, the better solutions are generated for the next iteration. Focusing on that, in this investigation, we use opposition based GA for discrete optimization problems.

The above proposed MPRPwR is formulated and numerically illustrated by the developed OBGA. Comparison are performed on the proposed OBGA using benchmark TSPLIB problems [9].

### 1.2 Literature Review

Traveling salesman problem (TSP) is an age-long problem and has several applications in real life areas. Real life application of TSP are observed in Pickup and Delivery [4], routing [2], traveler purchaser problems [7], etc. Recently, [5] investigated constrained solid TSPs in different environments (such as crisp, fuzzy, random, random-fuzzy, fuzzy-random and bi-random) and solved through GA.

Recently, [1] focused on the security for high-value shipment/transportation in the routing problem. They considered mixed integer linear programming for formulation of the problem with time window and dynamic risk index. [11] investigated cash-in-transit and risk threshold vehicle routing problem. They considered risk in terms of robbery depending on the distance and amount of cash carried. Also they express risk using occurrence of the event and the vulnerability. Similar type of another work on cash-in-transit and risk threshold was investigated by [12] and solved using ACO-LNS algorithm.

In the above formulations, none considered multiple routes between the destinations/places. In the present investigation, following [11], we have formulated MPRPwR with multiple paths and both path-dependent (obtained from past records) and path-goods weight dependent risks.

Since GA is a well established meta-heuristic approach to solve combinatorial optimization problem, we solve the proposed MPRPwR by a variant of GA. For more accurate findings and maintain diversity of solution space, we introduce opposition-based learning in GA. [10] focused on OBGA with opposition-based initialization with modified crossover for community detection problem. They only considered OB in initialization. In this paper, we impose OB in initialization as well as mutation process. [16] focused on OBACO for solving symmetric TSP. They also discussed on different strategies for pheromone update rules- direction, indirection, and random methods including ACO-Index, ACO-MaxIt, and ACO-Rand.

In this investigation, our proposed OBGAs, combine with probabilistic selection, comparison crossover and a generation dependent OB mutation. The models are illustrated numerically.

Novalities in this investigation are as follows:

- Opposition-based initialization is introduced in genetic algorithm
- Probabilistic selection procedure, comparison crossover and generation dependent OB mutation are introduced.

- Multi-path routing with risk is formulated and solved by OBGA.

The paper is organized as follows: Section 1 presents a concise introduction with motivation in section 1.1 and literature review in section 1.2. Section 2 explains details process of OBGA. The mathematical formulation of the problem is given in section 3. Section 4 used for computational experiments of the OBGA. In section 5, solution methodology is presented. Sections 6 and 7 represent numerical experiments and discussions respectively. Conclusion and future scope are presented in section 8.

# 2 Proposed opposition-based genetic algorithm (OBGA)

## 2.1 Representation

Let there are M paths covering N cities. For i$^{th}$ path, N-dimensional integer vectors $X_i = (x_{i1}, x_{i2}, \cdots, x_{iN})$ are created where $x_{i1}, x_{i2}, \cdots, x_{iN}$ indicate N consecutive nodes in a tour. Here, $x_{ij}$, i=1, 2, ⋯, M and j=1, 2, ⋯, N are randomly generated by a random number generator function between 1 to N maintaining the TSP conditions. Fitnesses of path are evaluated by summing the costs between the consecutive nodes of each solution (chromosome). The i$^{th}$ solution fitness in the solution space is presented by $f(X_i)$. As the population size is M, therefore M numbers of chromosomes are randomly generated.

## 2.2 Opposition-based learning

Till now, mainly opposition-based learning are used for solving continuous optimization problems. Opposition-based learning, basically implemented to make diversity in continuous search space, Focusing on that, we want to solve combinatorial/discrete optimization problems.

The opposition based concept in continuous domain, for D dimensional vectors ($R^D$), is first introduced by Tizhoosh (cf. [14]). Assume, $T = (y_1, y_2, \cdots, y_D)$ be any point in $R^D$, where $y_1, y_2, \cdots, y_D \in R$ and $y_i \in [a_i, b_i]$ , ∀ i ∈ $1, 2, \cdots, D$. The opposite point $T^c = (y_1^c, y_2^c, \cdots, y_D^c)$ defined as $y_i^c = a_i + b_i - y_i$. Here, the opposite point of y (in one dimensional) is shown in Fig. 1.

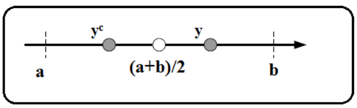



Figure 1: Example of opposition base point

### 2.3 Initialization

Using this approach in initialization shown in Fig. 2 of proposed algorithm let us consider, first node is fixed (depot), represented by 1 and initial population size=4. Then after using opposition-base learning, population size becomes 4+4=8. Now, we select best 4 out of 8 chromosomes using subsection 2.4.

| | | | | | |
|---|---|---|---|---|---|
| Chromosome 1 | 1 | 4 | 5 | 2 | 3 |
| Chromosome 2 | 1 | 5 | 2 | 3 | 4 |
| Chromosome 3 | 1 | 5 | 4 | 2 | 3 |
| Chromosome 4 | 1 | 5 | 2 | 3 | 4 |

| | | | | | |
|---|---|---|---|---|---|
| OB_Chromosome 1 | 1 | 2 | 5 | 4 | 3 |
| OB_Chromosome 2 | 1 | 5 | 4 | 3 | 2 |
| OB_Chromosome 3 | 1 | 5 | 2 | 4 | 3 |
| OB_Chromosome 4 | 1 | 5 | 4 | 3 | 2 |

| | | | | | |
|---|---|---|---|---|---|
| Chromosome 1 | 1 | 4 | 5 | 2 | 3 |
| Chromosome 2 | 1 | 5 | 2 | 3 | 4 |
| Chromosome 3 | 1 | 5 | 4 | 2 | 3 |
| Chromosome 4 | 1 | 5 | 2 | 3 | 4 |
| Chromosome 5 | 1 | 2 | 5 | 4 | 3 |
| Chromosome 6 | 1 | 5 | 4 | 3 | 2 |
| Chromosome 7 | 1 | 5 | 2 | 4 | 3 |
| Chromosome 8 | 1 | 5 | 4 | 3 | 2 |

Figure 2: Opposition base initialization

### 2.4 Selection (Probabilistic selection)

We first calculate the Boltzmann-Probability ( [5]) for each chromosome of the initial population using

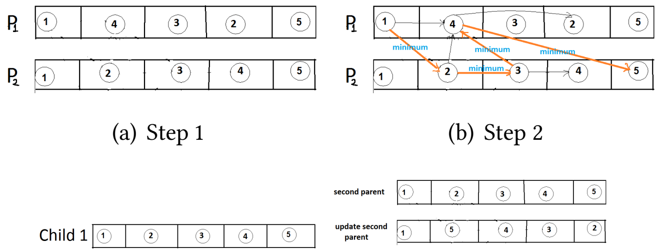
$$p_B = e^{((g/G)*(f_{min}-f(X_i))/KT)},$$

where, $T=T_0(1-a)^k$, $k = (1 + C * rand[0, 1])$, C=rand[1, 100], g=current generation number, $G$= max-gen, $T_0$= rand[50, 140], a=rand[0, 1], f($X_i$) is the objective function, $f_{min}$=min f($X_i$), i=1,2,$\cdots$, N.

For the mating pool, the following process is followed. First, a predefined value, say probability of selection ($p_s$) is assigned. If each chromosome of $f(X_i)$, a random number, r $\in$[0, 1] is generated. If $r < p_s$ or $r < p_B$, then the corresponding chromosome is selected for the mating pool. Otherwise, chromosomes corresponding to $f_{min}$ is selected for the mating pool. Finally, out of these chromosomes, 4 chromosomes are selected for crossover.

### 2.5 Crossover (comparison crossover)

Initially, two individuals (parents) are selected randomly from the mating pool, based on the random number generated between [0, 1]. Select the first parent (say $P_1$) according to $r < p_c$. Similarly, other parent (say $P_2$) is selected. Out of these parents, children are created using comparison crossover. The procedure to produce offspring is illustrated with an example for five node TSP (Fig. 3).




(a) Step 1 (b) Step 2

(c) Step 3 (d) Update the second parent

Figure 3: Comparison crossover for child 1 and 2

As we use comparison crossover and starting node of two parents are same (1) so, to generate second child, we update (reverse) the nodes of second parent as shown in Fig. 3(d). After that following same process for second child as first one.

## 2.6 Mutation

### 2.6.1 Generation dependent opposition based (OB) mutation

Here, we formulate a novel generation dependent OB mutation. Here generation based $p_m$ (probability of mutation) is evaluate as

$$p_m = \frac{t}{g},\ t \in (0, 1). \qquad \text{where, } g \text{ is the current generation number.}$$

### 2.6.2 Mutation process

If $r < p_m$, $r \in$ rand $[0, 1]$, then corresponding chromosome is selected for mutation. Opposition based mutation (except for $1^{st}$ and last node, as depot is fixed) shown in Algorithm 1 and Fig. 4.

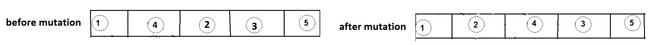



Figure 4: Mutation process

**Algorithm 1** Generation dependent OB mutation

**Step 1**: set g=current generation number
**Step 2**: $p_m=\frac{t}{g}$, t∈(0, 1).
**Step 3**: **for**(i=0; i< *population*; i++)
**Step 4**: $r = rand(0, 1)$
**Step 5**: **if**( $r < p_m$)
**Step 6**: choose current solution $a_i$, (i=1, ⋯, N)
**Step 7**: mutated solution $m_i$=(1+N)-$a_i$, (i=2, ⋯, N-1) (cf. Fig. 4)
**Step 8**: end if
**Step 9**: end for

After the selection, crossover and mutation procedures are obtained. If rand(0,1) is less than the opposition jumping rate (JR), then total population is the existing population (4) and created opposition based population (4) and from these 8 chromosomes, we select best 4 chromosomes as per subsection 2.4.

Combination of the above steps leads to the proposed OBGA presented in Algorithm 2.

**Algorithm 2** Opposition-based genetic algorithm (OBGA)

**Require**: pop size, jumping rate (JR), current generation (g) and maximum generation (G)
**Ensure**: optimum results
**Step 1**. start
**Step 2**. randomly generate initial population(M)
**Step 3**. opposite population calculate, using subsection 2.2 (M')
**Step 4**. g=1
**Step 5**. while ($g <= G$)
**Step 6**. selecting M fittest solutions from (M+M') using subsection 2.4
**Step 7**. comparison crossover using subsection 2.5
**Step 8**. generation dependent OB mutation according to Algorithm 1
**Step 9**. if rand $(0, 1) < JR$ then
**Step 10**. evaluate opposite population (M') of current population (M)
**Step 11**. g=g+1
**Step 12**. end if
**Step 13**. end while
**Step 14**. end

# 3 Proposed Multi-path Routing Problem with risk (MPRPwR)

Table 1: Notation and description of parameters and decision variables

| Notation | Description |
|---|---|
| N | number of nodes (1, 2, 3,⋯, N) |
| $i, j, k$ | index set |
| $Q$ | set of nodes {1, 2, 3,⋯, N}, $N = 1$ is depot |
| $P$ | set of routes $r_q \in \{1, 2, 3, \cdots\} = P$ |
| $x_i$ | $i^{th}$ visiting point |
| $x_{ij}$ | binary decision variable, $x_{ij} = 1$ for the travel from $i^{th}$ node to $j^{th}$ node, else, $x_{ij} = 0$ |
| $c(x_i, x_{i+1}, r_q)$ | traveling cost from $i^{th}$ node to $(i+1)^{th}$ node using $r_q \in P$ route per unit distance |
| $p(x_i, x_{i+1}, r_q)$ | probability of a robbery happening between $i^{th}$ and $(i+1)^{th}$ nodes using $r_q \in P$ route |
| $d_i$ | demand of materials at $i^{th}$ node |
| $dis(x_i, x_{i+1}, r_q)$ | distance between $i^{th}$ and $(i+1)^{th}$ nodes using $r_q \in P$ route |
| $D$ | total demand of collected materials |
| $\gamma(x_i, x_{i+1}, r_q)$ | vulnerability i.e, the probability of successful robbery between $i^{th}$ and $(i+1)^{th}$ nodes using $r_q \in P$ route |
| $L$ | per unit loading cost |
| $f(x_i, x_{i+1}, r_q)$ | fixed cost (toll tax) between $i^{th}$ and $(i+1)^{th}$ nodes using $r_q \in P$ route |
| $CR$ | cumulative risk of the entire route |
| $R_{max}$ | maximum permissible risk of the entire route |

## 3.1 Classical TSP (2DTSP)

Let $c(i, j)$ is the traveling cost from $i^{th}$ city to $j^{th}$ city. Then classical TSP (CTSP) can be mathematically represented as

$$\left.\text{Minimize} \quad Z = \sum_{i \neq j} c(i, j) x_{ij} \quad \right\} \tag{73.1}$$

$$\left.\begin{array}{l} \text{subject to} \quad \sum_{i=1}^{N} x_{ij} = 1 \ \text{ for } j = 1, 2, \cdots, N; \ \sum_{j=1}^{N} x_{ij} = 1 \ \text{ for } i = 1, 2, \cdots, N \\ \sum_{i \in S}^{N} \sum_{j \in S}^{N} x_{ij} \leq |S| - 1, \forall S \subset Q; \ \ where \ \ x_{ij} \in \{0, 1\}, i, j = 1, 2, \cdots, N. \end{array}\right\} \tag{73.2}$$

where Q={1, 2, 3,⋯, N} represents set of nodes, $x_{ij}$ are the decision variables. If the salesman travels from $i^{th}$ city to $j^{th}$ city, then $x_{ij} = 1$, else $x_{ij} = 0$. The first two constraints in Eq. 73.2 imply the visit of a node only once and the third constraint eliminates the sub route. Then the mentioned CTSP can be represented as

$$\left.\begin{array}{c} \text{determine a complete tour } (x_1, x_2, \cdots, x_N, x_1) \text{ to minimize} \\ Z = \sum_{i=1}^{N-1} c(x_i, x_{i+1}) + c(x_N, x_1) \\ \textit{where } i = 1, 2, \cdots, N. \textit{ and the constraints Eq. } 73.2. \end{array}\right\} \quad (73.3)$$

## 3.2 Multi-path TSP (MPTSP)

Here, we consider the availability of several paths connecting the cities. Let $c(x_i, x_{i+1}, r_q)$ is the traveling cost from $i^{th}$ city to $(i+1)^{th}$ city using $(r_q)^{th}$ route per unit distance. The salesman determines entire tour $(x_1, x_2, \cdots, x_N, x_1)$ using the routes $r_q \in \{1, 2, \cdots, P\}$ for the tour from $x_i$ to $x_{i+1}$, where $x_i \in \{1, 2, \cdots, N\}$ for $i = 1, 2, \cdots, N$, $r_q \in \{1, 2, \cdots, P\}$. Then this problem is mathematically represented as:

$$\left.\text{minimize } Z = \sum_{i=1}^{N-1} c(x_i, x_{i+1}, r_q) * dis(x_i, x_{i+1}, r_q) + c(x_N, x_1, r_q) * dis(x_N, x_1, r_q)\right\} \quad (73.4)$$

## 3.3 MPTSP with risk

A man purchases the items from some markets as per the demands. He starts the journey from the depot (first node, say) and following the TSP type tour scheme, travels to the market places (nodes) using one of the available routes at nodes and purchases the item.

### 3.3.1 Measuring the risk

The robbed risk between the $i^{th}$ and $j^{th}$ node using $r_q$ route can be measured as follows:

(1) $p(x_i, x_j, r_q)$ (denoted by $p_{i,j,r_q}$) is the probability of occurrence of robbery. It depends on various circumstances such as available paths, type of vehicle, road condition, weather condition, time (day/night), etc.

(2) The vulnerability $\gamma(x_i, x_j, r_q)$ (denoted by $\gamma_{i,j,r_q}$) is the probability of successful robbery. It also depends on various circumstances such as efficiency of thieves, activeness of police, etc.

(3) The risk is also depends on the length $(dis(x_i, x_j, r_q))$ of the edge traversed by the vehicle

(4) It also depends on the amount of collected materials $(d_i)$.

For the present model, two types of probability of robbery are considered. First we assume that robbery, depends on length ($dis(x_i, x_j, r_q)$) of the edge and collected materials ($d_i$). So $p_{i,j,r_q}$ is evaluated based on distance and pick up materials. For the second case, this robbery probability is taken from the path wise past records, independent of distance and materials. The successful robbery parameter, ($\gamma_{i,j,r_q}$) is also taken from past records of that area.

#### 3.3.2 Risk at each arc with certain probability

As the chance of occurring the robbery, etc. is not certain, so risk can be expressed in terms of probability. Fig. 5 represents the overview of probabilistic risk.

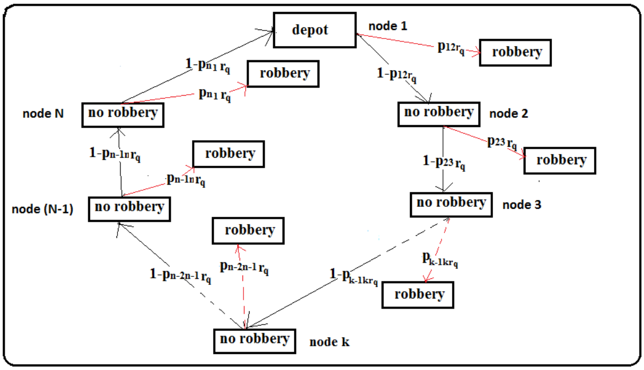



Figure 5: Graphical representation of probabilistic refusal

In this investigation probabilities of risk, $p_{i,j,r_q}(dis_{i,j,r_q}, (d_1, d_2, \cdots, d_N))$, $i, j = 1, 2, \cdots, N$ can be assigned to the arc. Here, $1^{st}$ node represent the depot. Risk in the arc 1-2 (i.e, between $1^{st}$ and $2^{nd}$ node with route $r_q$) with probability $p_{1,2,r_q}(dis_{1,2,r_q}, d_1)$ is,

$$= p_{1,2,r_q}(dis_{1,2,r_q}, d_1) * \gamma_{1,2,r_q}$$

Risk in the arc 2-3 via 1-2-3 (i.e, among $2^{nd}$ and $3^{rd}$ node with route $r_q$)) with probability $p_{2,3,r_q}(dis_{2,3,r_q}, (d_1, d_2))$ is,

$$=(1\text{-}p_{1,2,r_q}(dis_{1,2,r_q}, d_1)) * p_{2,3,r_q}(dis_{2,3,r_q}, (d_1, d_2)) * \gamma_{2,3,r_q}$$

Thus, Risk in the arc n-1 via 1-2-3-⋯-n-1(i.e, among $n^{th}$ and $1^{st}$ node with route $r_q$) with probability $p_{n,1,r_q}(dis_{n,1,r_q}, (d_1, d_2, \cdots, d_n))$ is,

$$= (1\text{-}p_{1,2,r_q}(dis_{1,2,r_q}, d_1))^*(1\text{-}p_{2,3,r_q}(dis_{2,3,r_q}, (d_1, d_2)))^* \cdots$$
$$^* (1\text{-}p_{n-1,n,r_q}(dis_{n-1,n,r_q}, (d_1, d_2, \cdots, d_{n-1}))) *p_{n,1,r_q}(dis_{n,1,r_q}, (d_1, d_2, \cdots, d_n)) * \gamma_{n,1,r_q}$$

**3.3.2.1 Model A: Risk depends on distance and collected materials** The model mathematically formulated as:

$$\left.\begin{aligned}\text{Min}\quad Z = \sum_{i=1}^{N-1} c(x_i, x_{i+1}, r_q) * dis(x_i, x_{i+1}, r_q) + c(x_N, x_1, r_q) * dis(x_N, x_1, r_q)\\ + \sum_{i=2}^{N} (d_i * L) + \sum_{i=1}^{N-1} f(x_i, x_{i+1}, r_q) * \xi(x_i, x_{i+1}, r_q) +\\ f(x_N, x_1, r_q) * \xi(x_N, x_1, r_q)\end{aligned}\right\} \quad (73.5)$$

$$\text{subject to}\quad CR = \sum_{i=1}^{N-1} R(x_i, x_{i+1}, r_q, d_i) + R(x_N, x_1, r_q, d_i) < R_{max} \Big\} \quad (73.6)$$

$$where\ \ r_q \in \{1, 2 \cdots, P\}\ \ \sum_{i=1}^{N} d_i = D,\ \ d_1 = 0 \Big\} \quad (73.7)$$

where, values of $R(x_i, x_{i+1}, r_q, d_i)$ presented in Table 2

Table 2: Values of $R(x_i, x_{i+1}, r_q, d_i)$

| weight (kg)/Distance (km) | 0≤dis< $l_1$ | $l_1$ ≤dis< $l_2$ | $l_2$ ≤dis< $l_3$ |
|---|---|---|---|
| $d_0 \leq d_i < d_1$ | $R_4$ | $R_7$ | $R_{10}$ |
| $d_1 \leq d_i < d_2$ | $R_5$ | $R_8$ | $R_{11}$ |
| $d_2 \leq d_i < d_3$ | $R_6$ | $R_9$ | $R_{12}$ |

(Here, $d_0, d_1, d_2, d_3, d_4$ are amount of collected materials, $l_1, l_2, l_3$ are length/ distance, $R_1, R_2, R_3, \cdots, R_{12}$ are risk)
Here total risk (CR) cumulatively increases during travels along the routes. $R_{max}$ is the maximum acceptable risk.

Here in Eq. 73.5 objective function Z has three parts, first part indicates the traveling cost, second part the loading cost of the precious items, and third part for fixed charge though out the routing. Eq. 73.6 represents risk constraints ($R_{max}$) which is used for feasible path. Total demand (D) is equal to the cumulative collection of materials from all nodes ($\sum_{i=1}^{N} d_i$) and first node is fixed for depot ($d_1$=0). For clarity, the proposed MPRPwR is illustrated in Fig. 6, with data from Table 7.

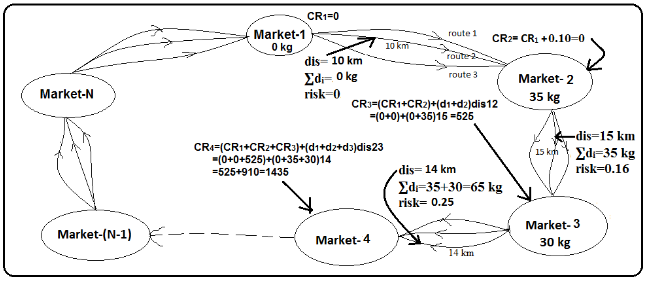



Figure 6: Graphical representation of model

**3.3.2.2 Model B: Risk independent of distance and collected materials** In the case, the model mathematically is formulated as:
Objective function is same as Model A (given in Eq. 73.5).

$$\left.\begin{aligned} \text{subject to } CR = \sum_{i=1}^{N-1} R(x_i, x_{i+1}, r_q) * \gamma(x_i, x_{i+1}, r_q) + R(x_N, x_1, r_q) * \\ \gamma(x_N, x_1, r_q) < R_{max} \;\; along \;\; with \;\; Eq.73.7 \end{aligned}\right\} \tag{73.8}$$

# 4 Computational Experiment

We design the algorithm in C/C++ using the Codeblock compiler on a $5^{th}$ Gen. Intel Core i3 CPU @ 3 GHz. The values of the parameter are given in Table 3.

Table 3: Parameters values chosen for numerical experiment

| Parameters | Value/Range | Parameters | Value/Range |
|---|---|---|---|
| Max_Generation | 200–1000 | $D$ | 152 |
| Number of Chromosome | 50–150 | $p_m$ | 0.31 |
| $p_c$ | 0.49 | $R_{max}$(Model A) | 3.15 |
| $JR$ | 0.32 | $R_{max}$(Model B) | 3.75 |
| $L$ | INR 30 | | |

## 4.1 Test results for OBGA

The performance of the intended OBGA is established by solving 20 standard benchmark problems from TSPLIB [ [8]]. Table 4 represents the results of OBGA. Comparing all the problems in terms of the total cost, number of iteration and cpu time. All the results are obtained using 25 independent runs and taken using two different algorithms- OBGA and GA (with Roulette Wheel selection, Cyclic crossover, Random mutation). In the Table 4, "BKS" indicates the best-known solution, and "BFS" indicates the best-found solution.

Table 4: Results for Standard TSP Problem (TSPLIB)

| Instances | BKS | OBGA | | | GA | | |
|---|---|---|---|---|---|---|---|
| | | BFS | Iteration | Time | BFS | Iteration | Time |
| us16 | 6859 | 6859 | 56 | 0.04 | 6859 | 64 | 0.07 |
| gr17 | 2085 | 2085 | 68 | 0.06 | 2085 | 79 | 0.14 |
| gr21 | 2707 | 2707 | 159 | 0.09 | 2707 | 187 | 0.19 |
| fri26 | 937 | 937 | 147 | 0.14 | 937 | 201 | 0.27 |
| bays29 | 2020 | 2020 | 134 | 0.19 | 2020 | 157 | 0.54 |
| dantzig42 | 699 | 699 | 245 | 0.35 | 699 | 270 | 0.87 |
| eil51 | 426 | 426 | 298 | 0.65 | 426 | 367 | 1.34 |
| berlin52 | 7542 | 7542 | 347 | 0.87 | 7687 | 407 | 1.57 |
| st70 | 675 | 675 | 462 | 1.35 | 697 | 490 | 2.34 |
| eil76 | 538 | 538 | 517 | 1.86 | 578 | 547 | 2.89 |
| pr76 | 108159 | 108159 | 623 | 2.14 | 109295 | 779 | 3.07 |
| rat99 | 1211 | 1211 | 748 | 2.71 | 1258 | 864 | 3.58 |
| eil101 | 629 | 629 | 855 | 3.38 | 638 | 1143 | 3.96 |
| kroA100 | 21282 | 21282 | 774 | 3.75 | 21618 | 1242 | 4.60 |
| kroC100 | 20749 | 20901 | 896 | 4.71 | 21354 | 1329 | 5.94 |
| kroA150 | 26524 | 26823 | 1167 | 5.04 | 26982 | 1468 | 7.29 |
| kroB200 | 29437 | 29635 | 1334 | 5.76 | 29913 | 1614 | 8.37 |
| a280 | 2579 | 2647 | 1469 | 7.27 | 27413 | 1795 | 10.25 |
| pr299 | 48191 | 50875 | 1622 | 8.43 | 52743 | 1828 | 11.49 |
| lin318 | 42029 | 43146 | 1985 | 8.57 | 45413 | 1997 | 11.62 |

# 5 Solutions of MPRPwR using OBGA

## 5.1 Solution methodology

The proposed algorithm OBGA is the combination of probabilistic selection, comparison crossover, and generation dependent OB mutation which was implemented in C++ with 150 chromosomes and 1000 iterations in maximum. The proposed MPRPwR are solved and numerical illustrated by the newly implemented OBGA for some input data.

# 6 Numerical experiments

## 6.1 Input data

For experiment, we consider MPRPwR with 10 nodes (places) and 3 alternative routes between every two nodes. The distance, traveling cost per unit distance, fixed charge cost, probability of risk and vulnerability matrices (Row-wise $1^{st}$, $2^{nd}$, $3^{rd}$, $4^{th}$ and $5^{th}$ sets correspond to distances, traveling cost per unit distance, fixed charge cost, probability of risk and vulnerability) for MPRPwR are presented in Table 5.

As mentioned earlier, there are three types of routes between two nodes with the corresponding distance and traveling cost matrices along different routes for the models. For the distance and traveling cost (a,b,c)(represented by (≡(0,1,2))) (say), the values a, b and c are for the $1^{st}$, $2^{nd}$ and $3^{rd}$ routes respectively. Demand (purchased amount) matrix (in weight) of materials is given in Table 6.

## 6.2 Routing Problems (Models-A and B) having single path between the nodes

Instead of multi-paths, we assume that there is only one route for travel between the nodes. Taking a particular path ('0' in this case); the routing problem is solved and results are given in Tables-8, and 9.

Table 6: Demand matrix

| i/j | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Demand (kg) | 0 | 10 | 15 | 20 | 22 | 16 | 18 | 24 | 12 | 15 |

Table 7: Distance and collected weight dependent risk

| weight (kg)/Distance (km) | 0≤dis<20 | 20≤dis<40 | 40≤dis<60 |
|---|---|---|---|
| $0 \leq d_i < 38$ | 0.08 | 0.16 | 0.25 |
| $38 \leq d_i < 76$ | 0.16 | 0.33 | 0.50 |
| $76 \leq d_i < 114$ | 0.25 | 0.50 | 0.75 |
| $114 \leq d_i < 152$ | 0.33 | 0.66 | 0.99 |

### 6.3 Optimum results of MPRPwR (different Models)

Table 8: Results of models A and B

| | | with risk | | without risk | |
|---|---|---|---|---|---|
| | parameters | with muti-path | with single path | with muti-path | with single path |
| Model A | path | 1(0)-5(2)-9(0)-4(0) -3(0)-7(2)-8(2)-10(2) -2(0)-6(0)-1 | 1(0)-5(0)-10(0)-2(0) -3(0)-7(0)-8(0)-6(0) -4(0)-9(0)-1 | 1(1)-6(0)-4(0)-5(0) -9(0)-8(2)-7(1)-3(0) -2(0)-10(2)-1 | 1(0)-5(0)-9(0)-4(0) -3(0)-7(0)-8(0)-6(0) -2(0)-10(0)-1 |
| | distance(km) | 246 | 260 | 198 | 202 |
| | traveling cost(INR) | 2044 | 2198 | 1597 | 1712 |
| | loading cost(INR) | 4110 | 4110 | 4110 | 4110 |
| | fixed charge(INR) | 83 | 87 | 82 | 87 |
| | total cost(INR) | 6237 | 6395 | 5789 | 5909 |
| | risk | 3.15 | 3.15 | 3.71 | 3.75 |
| Model B | path | 1(2)-6(2)-5(2)-3(0)-7(1)-4(1) -2(1)-10(2)-9(2)-8(2)-1 | 1(0)-2(0)-10(0)-8(0)-3(0)-7(0) -9(0)-4(0)-5(0)-6(0)-1 | | |
| | distance(km) | 213 | 249 | | |
| | traveling cost(INR) | 2173 | 2411 | | |
| | loading cost(INR) | 4110 | 4110 | -do- | |
| | fixed charge(INR) | 201 | 297 | | |
| | total cost(INR) | 6484 | 6818 | | |
| | risk | 1.79 | 2.46 | | |

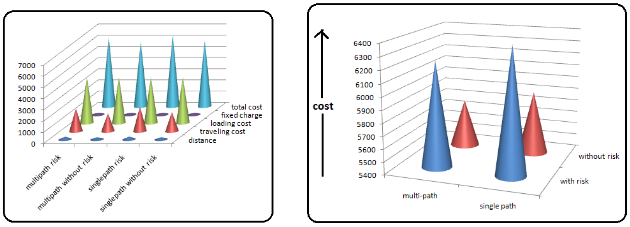
Table 9: Results of Model A and B using multi-path and without multi-path result

| Model A | | | Model B | | |
|---|---|---|---|---|---|
| risk limit | total cost (with multi-path) | total cost (with single path) | risk limit | total cost (with multi-path) | total cost (with single path) |
| 3.9 | 5789 (same) | 5909 (same) | 3.4 | 5789(same) | 5909(same) |
| 3.8 | 5789 (same) | 5909 (same) | 3.2 and above | 5789(same) | 5909(same) |
| 3.7 | 5893 | 5947 | 3.0 | 5789 (same) | 6036 |
| 3.6 | 5910 | 5960 | 2.8 | 5916 | 6360 |
| 3.5 | 6020 | 6072 | 2.6 | 5983 | 6703 |
| 3.4 | 6041 | 6087 | 2.4 | 5994 | 7324 |
| 3.3 | 6076 | 6395 | 2.2 | 6095 | no feasible path |
| 3.2 | 6237 | 6395 | 2.0 | 6119 | no feasible path |
| 3.1 | no feasible solution | no feasible solution | 1.8 | 6484 | no feasible path |
| | | | 1.7 and below | no feasible path | no feasible path |

## 7 Discussion

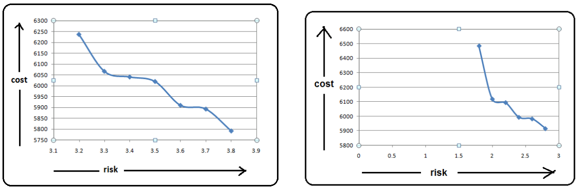
Tables 8 and Fig. 7(a) represent the overall scenario of Model A. It is observed that use of multi-path between the nodes gives less risk for the entire path, the system cost is also less than the single path model. Same results are found for the without risk cases (Fig. 7(b)).

In Table 8, the multi-path with risk is 1(0)-5(2)-9(0)-4(0)-3(0)-7(2)-8(2)-10(2)-2(0)-6(0)-1. The distance, traveling cost, loading cost, fixed charge, total cost and




(a) Models with all parameters (b) Cost of multi-path and single path

Figure 7: Overview of models




(a) Model A (b) Model B

Figure 8: Graphical representation of risk and cost

risk are 246 km, INR 2044, INR 4110, INR 83, INR 6237 and 3.15 respectively.

Table 9 represents the total system cost with different risks. It is seen that when risk limit decreases, the system cost increases (Fig. 8(a)), but when risk limit is less than 3.1, it is unable to find feasible path in the system. Again when risk increases, the system cost is minimized upto certain level (3.8). When risk is more above than 3.8, the system cost remains unaltered (Table 9). This behavior is as per expectation.

Similar scenarios are observed for Model B (Table 8, 9). Also for this model, when risk limit decreases, the system cost increases (Fig. 8(b)), but when risk limit is less than 1.79, we don't get any feasible path in the system. Again when risk is increased, the system cost is minimized upto certain level (2.83). When risk is above than 2.83, the system cost remains unaltered.

Table 9 also furnishes that multi-path routing gives feasible paths with much lower risks, which is not possible for the single path routing model.

## 8 Conclusion and future scope

In this paper, opposition based genetic algorithm (OBGA) with (i) probabilistic selection, (ii) comparison crossover and (iii) Generation Dependent OB mutation are proposed and implemented successfully in some MPRPwR problems. In routing problems, we consider (i) risk constraint (ii) fixed charge (toll tax), (iii) loading charge for the items. Here, two types of risk are considered along the routes-i.e risk (i) depends on distance and amount of collected materials, (ii) depends on the path only (from past records). This formulation can be used in the fields of inventory control, supply-chain etc., for better managerial decision making.

The limitations of our investigation are that we consider hypothetical input data for risk and vulnerability. In future, these models can be extended considering multi vehicles along with multi path among the nodes. Here OB is used in initialization and mutation. OB can also be used in selection and crossover for different versions of GA.

Table 5: Input Data: Distance, traveling cost per unit distance and fixed charge cost, probability of risk, vulnerability matrix

| i/j | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ∞ | (29,27,33) | (32,30,27) | (37,34,32) | (18,26,21) | (18,16,14) | (37,40,35) | ( 39,35,32) | (42,47,44) | (32,29,26) |
| | ∞ | (10,8,11) | (9,8,11) | (8,9,10) | (9,7,11) | (8,9,10) | (11,9,8) | (9,8,11) | (7,9,11) | (9,10,7) |
| | ∞ | (110,111,118) | (0,0,0) | (8,9,10) | (9,7,11) | (8,9,10) | (11,9,8) | (9,8,11) | (7,9,11) | (9,10,7) |
| | ∞ | (.29,.27,.33) | (.32,30,27) | (.37,.34,.32) | (.18,.26,.21) | (.18,.16,.14) | (.37,.40,.35) | (.39,.35,.32) | (.42,.47,.44) | (.32,.29,.26) |
| | ∞ | (.10,.8,.11) | (.9,.8,.11) | (.8,.9,.10) | (.9,.7,.11) | (.8,.9,.10) | (.11,.9,.8) | (.9,.8,.11) | (.7,.9,.11) | (.9,.10,.7) |
| 2 | (29,27,33) | ∞ | (22,26,28) | (30,26,31) | (37,36,38) | (48,46,43) | (32,36,33) | (44,46,47) | (30,26,25) | (16,15,17) |
| | (10,11,8) | ∞ | (8,9,8) | (9,10,11) | (8,10,9) | (7,9,10) | (11,10,9) | (8,9,8) | (7,10,8) | (11,10,9) |
| | (0,0,0) | ∞ | (8,9,118) | (9,10,11) | (8,10,9) | (7,9,10) | (11,10,9) | (8,9,8) | (117,10,8) | (11,10,9) |
| | (.29,.27,.33) | ∞ | (.22,.26,.28) | (.30,.26,.31) | (.37,.36,.38) | (.48,.46,.43) | (.32,.36,.33) | (.44,.46,.47) | (.30,.26,.25) | (.16,.15,.17) |
| | (.10,.11,.8) | ∞ | (.8,.9,.8) | (.9,.10,.11) | (.8,.10,.9) | (.7,.9,.10) | (.11,.10,.9) | (.8,.9,.8) | (.7,.10,.8) | (.11,.10,.9) |
| 3 | (32,30,27) | (22,26,28) | ∞ | (32,36,39) | (27,26,23) | (37,39,41) | (11,16,15) | (34,38,35) | (42,44,47) | (32,37,34) |
| | (9,8,11) | (8,9,8) | ∞ | (8,10,10) | (9,7,11) | (10,9,8) | (11,9,9) | (10,9,10) | (8,8,8) | (9,8,10) |
| | (9,118,11) | (8,9,8) | ∞ | (8,10,10) | (9,7,11) | (10,9,8) | (11, 119,9) | (10,9,10) | (8,8,8) | (9,8,10) |
| | (.32,.30,.27) | (.22,.26,.28) | ∞ | (.32,.36,.39) | (.27,.26,.23) | (.37,.39,.41) | (.11,.16,.15) | (.34,.38,.35) | (.42,.44,.47) | (.32,.37,.34) |
| | (.9,.8,.11) | (.8,.9,.8) | ∞ | (.8,.10,.10) | (.9,.7,.11) | (10,.9,.8) | (.11,.9,.9) | (.10,.9,.10) | (.8,.8,.8) | (.9,.8,.10) |
| 4 | (37,34,32) | (30,26,31) | (32,36,39) | ∞ | (22,26,28) | (27,26,28) | (32,36,35) | (37,36,40) | (24,26,22) | (39,36,34) |
| | (8,9,10) | (9,10,11) | (8,10,10) | ∞ | (9,8,10) | (7,9,9) | (8,10,9) | (9,9,10) | (8,9,10) | (9,8,9) |
| | (8,0,10) | (9,10,11) | (8,10,10) | ∞ | (9,8,10) | (7,9,9) | (8,10,9) | (0,119,10) | (0,0,0) | (9,8,9) |
| | (.37,.34,.32) | (.30,.26,.31) | (.32,.36,.39) | ∞ | (.22,.26,.28) | (.27,.26,.28) | (.32,.36,.35) | (.37,.36,.40) | (.24,.26,.22) | (.39,.36,.34) |
| | (.8,.9,.10) | (.9,.10,.11) | (.8,.10,.10) | ∞ | (.9,.8,.10) | (.7,.9,.9) | (.8,.10,.9) | (.9,.9,.10) | (.8,.9,.10) | (.9,.8,.9) |
| 5 | (18,26,21) | (37,36,38) | (27,26,23) | (22,26,28) | ∞ | (22,26,24) | (23,26,28) | (32,36,38) | (16,16,18) | (30,36,31) |
| | (9,7,11) | (8,9,10) | (9,7,11) | (9,8,10) | ∞ | (10,9,10) | (9,8,7) | (9,10,8) | (7,9,9) | (8,10,9) |
| | (9,0,11) | (8,9,10) | (9,7,11) | (9,8,10) | ∞ | (10,9,10) | (9,0,7) | (9,10,8) | (7,112,9) | (8,10,9) |
| | (.18,.26,.21) | (.37,.36,.38) | (.27,.26,.23) | (.22,.26,.28) | ∞ | (.22,.26,.24) | (.23,.26,.28) | (.32,.36,.38) | (.16,.16,.18) | (.30,.36,.31) |
| | (.9,.7,.11) | (.8,.9,.10) | (.9,.7,.11) | (.9,.8,.10) | ∞ | (.10,.9,.10) | (.9,.8,.7) | (.9,.10,.8) | (.7,.9,.9) | (.8,.10,.9) |
| 6 | (18,16,14) | (48,46,43) | (37,39,41) | (27,26,28) | (22,26,24) | ∞ | (34,36,38) | (42,46,47) | (34,36,38) | (32,36,30) |
| | (8,9,10) | (7,9,10) | (10,9,8) | (7,9,9) | (10,9,10) | ∞ | (7,8,8) | (9,9,11) | (10,9,8) | (8,9,9) |
| | (8,9,110) | (7,119,10) | (0,0,8) | (7,9,0) | (10,9,10) | ∞ | (7,8,8) | (0,0,0) | (10,9,8) | (8,9,9) |
| | (.18,.16,.14) | (.48,.46,.43) | (.37,.39,.41) | (.27,.26,.28) | (.22,.26,.24) | ∞ | (.34,.36,.38) | (.42,.46,.47) | (.34,.36,.38) | (.32,.36,.30) |
| | (.8,.9,.10) | (.7,.9,.10) | (.10,.9,.8) | (.7,.9,.9) | (.10,.9,.10) | ∞ | (.7,.8,.8) | (.9,.9,.11) | (.10,.9,.8) | (.8,.9,.9) |
| 7 | (37,40,35) | (32,36,33) | (11,16,15) | (32,36,35) | (23,26,28) | (34,36,38) | ∞ | (12,16,13) | (34,36,32) | (22,26,20) |
| | (11,9,8) | (11,10,9) | (11,9,9) | (8,10,9) | (9,8,7) | (7,8,8) | ∞ | (8,9,8) | (10,10,9) | (11,9,8) |
| | (11,110,8) | (0,10,9) | (111,9,9) | (8,10,9) | (9,0,7) | (7,8,8) | ∞ | (8,9,8) | (10,10,9) | (11,0,8) |
| | (.37,.40,.35) | (.32,.36,.33) | (.11,.16,.15) | (.32,.36,.35) | (.23,.26,.28) | (.34,.36,.38) | ∞ | (.12,.16,.13) | (.34,.36,.32) | (.22,.26,.20) |
| | (.11,.9,.8) | (.11,.10,.9) | (.11,.9,.9) | (.8,.10,.9) | (.9,.8,.7) | (.7,.8,.8) | ∞ | (.8,.9,.8) | (.10,.10,.9) | (.11,.9,.8) |
| 8 | (39,35,32) | (44,46,47) | (34,38,35) | (37,36,40) | (32,36,38) | (42,46,47) | (12,16,13) | ∞ | (24,26,25) | (39,36,42) |
| | (9,8,11) | (8,9,8) | (10,9,10) | (9,9,10) | (9,10,8) | (9,9,11) | (8,9,8) | ∞ | (9,8,10) | (10,9,11) |
| | (9,0,0) | (0,0,118) | (10,9,10) | (9,9,10) | (0,10,8) | (9,9,11) | (8,9,8) | ∞ | (9,8,10) | (10,9,11) |
| | (.39,.35,.32) | (.44,.46,.47) | (.34,.38,.35) | (.37,.36,.40) | (.32,.36,.38) | (.42,.46,.47) | (.12,.16,.13) | ∞ | (.24,.26,.25) | (.39,.36,.42) |
| | (.9,.8,.11) | (.8,.9,.8) | (.10,.9,.10) | (.9,.9,.10) | (.9,.10,.8) | (.9,.9,.11) | (.8,.9,.8) | ∞ | (.9,.8,.10) | (.10,.9,.11) |
| 9 | (42,47,44) | (30,26,25) | (42,44,47) | (24,26,22) | (16,16,16) | (34,36,38) | (34,36,32) | (24,26,25) | ∞ | (30,26,25) |
| | (7,9,11) | (7,10,8) | (8,8,8) | (8,9,10) | (7,9,9) | (10,9,8) | (10,10,9) | (9,8,10) | ∞ | (9,11,10) |
| | (7,9,11) | (7,10,8) | (8,8,8) | (8,0,10) | (7,9,9) | (10,9,8) | (10,10,9) | (9,8,10) | ∞ | (9,11,10) |
| | (.42,.47,.44) | (.30,.26,.25) | (.42,.44,.47) | (.24,.26,.22) | (.16,.16,.16) | (.34,.36,.38) | (.34,.36,.32) | (.24,.26,.25) | ∞ | (.30,.26,.25) |
| | (.7,.9,.11) | (.7,.10,.8) | (.8,.8,.8) | (.8,.9,.10) | (.7,.9,.9) | (.10,.9,.8) | (.10,.10,.9) | (.9,.8,.10) | ∞ | (.9,.11,.10) |
| 10 | (41,29,26) | (16,15,17) | (32,37,34) | (39,36,34) | (30,36,31) | (32,36,30) | (22,26,20) | (39,36,42) | (30,26,25) | ∞ |
| | (9,10,7) | (11,10,9) | (9,8,10) | (9,8,9) | (8,10,9) | (8,9,9) | (11,9,8) | (10,9,11) | (9,11,10) | ∞ |
| | (9,10,7) | (11,10,9) | (119,0,10) | (9,8,9) | (8,10,9) | (0,9,9) | (111,9,8) | (110,9,11) | (9,111,10) | ∞ |
| | (.41,.29,.26) | (.16,.15,.17) | (.32,.37,.34) | (.39,.36,.34) | (.30,.36,.31) | (.32,.36,.30) | (.22,.26,.20) | (.39,.36,.42) | (.30,.26,.25) | ∞ |
| | (.9,.10,.7) | (.11,.10,.9) | (.9,.8,.10) | (.9,.8,.9) | (.8,.10,.9) | (.8,.9,.9) | (.11,.9,.8) | (.10,.9,.11) | (.9,.11,.10) | ∞ |

# References

[1] Allahyari, S., Yaghoubi, S., and Van Woensel, T. (2021). A novel risk perspective on location-routing planning: An application in cash transportation. *Transportation Research Part E: Logistics and Transportation Review*, 150:102356.

[2] Bellman, R. (1958). On a routing problem. *Quarterly of applied mathematics*, 16(1):87–90.

[3] Goeke, D. (2019). Granular tabu search for the pickup and delivery problem with time windows and electric vehicles. *European Journal of Operational Research*, 278(3):821–836.

[4] Kalantari, B., Hill, A. V., and Arora, S. R. (1985). An algorithm for the traveling salesman problem with pickup and delivery customers. *European Journal of Operational Research*, 22(3):377–386.

[5] Maity, S., Roy, A., and Maiti, M. (2015). A modified genetic algorithm for solving uncertain constrained solid travelling salesman problems. *Computers & Industrial Engineering*, 83:273–296.

[6] Malik, H. I. (2021). *Determinants of Foreign Direct Investment; Evidence From Asian Countries*. PhD thesis, CAPITAL UNIVERSITY.

[7] Manerba, D., Mansini, R., and Riera-Ledesma, J. (2017). The traveling purchaser problem and its variants. *European Journal of Operational Research*, 259(1):1–18.

[8] Reinelt, G. (1991). Tsplib—a traveling salesman problem library. *ORSA journal on computing*, 3(4):376–384.

[9] Reinelt, G. (1995). Tsplib http://www. iwr. uni-heidelberg. de/groups/comopt/software. *TSPLIB95*.

[10] Shakya, H. K., Singh, K., More, Y. S., and Biswas, B. (2020). Opposition-based genetic algorithm for community detection in social networks. *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*, pages 1–13.

2021. In Rahul Srivastava & Aditya Kumar Singh Pundir (eds.), *New Frontiers in Communication and Intelligent Systems*, 727–746. Computing & Intelligent Systems, SCRS, India. https://doi.org/10.52458/978-81-95502-00-4-73

[11] Talarico, L., Sörensen, K., and Springael, J. (2015). Metaheuristics for the risk-constrained cash-in-transit vehicle routing problem. *European Journal of operational research*, 244(2):457–470.

[12] Talarico, L., Springael, J., Sörensen, K., and Talarico, F. (2017). A large neighbourhood metaheuristic for the risk-constrained cash-in-transit vehicle routing problem. *Computers & Operations Research*, 78:547–556.

[13] Tikani, H., Setak, M., and Demir, E. (2021). A risk-constrained time-dependent cash-in-transit routing problem in multigraph under uncertainty. *European Journal of Operational Research*, 293(2):703–730.

[14] Tizhoosh, H. R. (2005). Opposition-based learning: a new scheme for machine intelligence. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, volume 1, pages 695–701. IEEE.

[15] Wang, H., Wu, Z., and Rahnamayan, S. (2011). Enhanced opposition-based differential evolution for solving high-dimensional continuous optimization problems. *Soft Computing*, 15(11):2127–2140.

[16] Zhang, Z., Xu, Z., Luan, S., Li, X., and Sun, Y. (2020). Opposition-based ant colony optimization algorithm for the traveling salesman problem. *Mathematics*, 8(10):1650.