Soft Comput
DOI 10.1007/s00500-016-2075-4

METHODOLOGIES AND APPLICATION

A genetic ant colony optimization based algorithm for solid multiple travelling salesmen problem in fuzzy rough environment

Chiranjit Changdar[1] · Rajat Kumar Pal[2] · G. S. Mahapatra[3]

Abstract In this paper, a genetic-ant colony optimization algorithm has been presented to solve a solid multiple Travelling Salesmen Problem (mTSP) in fuzzy rough environment. In solid mTSP, a set of nodes (locations/cities) are given, and each of the cities must be visited exactly once by the salesmen such that all of them start and finish at a depot using different conveyance facility. A solid mTSP is an extension of mTSP where the travellers use different conveyance facilities for travelling from one city to another. To solve an mTSP, a hybrid algorithm has been developed based on the concept of two algorithms, namely genetic algorithm (GA) and ant colony optimization (ACO) based algorithm. Each salesman selects his/her route using ACO and the routes of different salesmen (to construct a complete solution) are controlled by the GA. Here, a set of simple ACO characteristics have further been modified by incorporating a special feature namely 'refinement'. In this paper, we have utilized cyclic crossover and two-point's mutation in the proposed algorithm to solve the problem. The travelling cost is considered as imprecise in nature (fuzzy-rough) and is reduced to its approximate crisp using fuzzy-rough expectation. Computational results with different data sets are presented and some sensitivity analysis has also been made.

Keywords Solid multiple travelling salesman problem · Ant colony optimization · Genetic algorithm · Refinement · Cyclic crossover

Communicated by V. Loia.

✉ Chiranjit Changdar
chiranjit_changdar@yahoo.co.in

Rajat Kumar Pal
pal.rajatk@gmail.com

G. S. Mahapatra
g_s_mahapatra@yahoo.com

1 Department of Computer Science, Raja N.L. Khan Women's College, Midnapore 721102, West Bengal, India

2 Department of Computer Science and Engineering, University of Calcutta, Calcutta 700 098, West Bengal, India

3 Department of Mathematics, National Institute of Technology, Puducherry 609605, Karaikal, India

1 Introduction

A multiple Travelling Salesman Problem (mTSP) is another type of well-known travelling salesman problem (TSP). In the present market situation, mTSP is more suitable when it is compared to the classical TSP for modelling different practical situations. It has a capability of handling two or more salesmen. In case of single depot mTSP, a set of nodes (locations/cities) are given and all of the cities are to be visited exactly once by a salesman such that all of them start and finish at the single depot node. In this mTSP, there are a given number of cities and a particular number of salesmen. All the salesmen start to visit the cities from the depot node. Each salesman has to visit at least a minimum number of cities and do not allow to visit greater than a maximum number of cities. The goal is to find the tour schedule of each salesman such that the total travelling cost of all salesmen becomes minimum. The cost matrix is defined in terms of distance, time, and so on. A solid mTSP has been presented where different conveyance facilities are available to travel from city to city. Due to different conveyance facilities, the problem mTSP is termed as 'solid' mTSP.

However, many practical problems have been modelled using multiple TSP, such as crew scheduling (Bigras et al. 2008), cold rolling scheduling (Zhao et al. 2011), mission

Published online: 21 March 2016

planning (Ryan et al. 1998), and so on. Multiple TSP with precedence constraints has been explained by Sarin et al. (2014). The mTSP can be generalized to a wide variety of routing and scheduling problems, for example, the school bus routing problem (Orloff 1974) and the pickup and delivery problem (Mladenović et al. 2012; Salazar-González and Santos-Hernández 2015). An mTSP has been modelled by Kiraly et al. (2015) for minimization of off-grade production in multi-site multi-product plants.

The multiple travelling salesman problem (mTSP) is a generalization of the well-known travelling salesman problem. In the past two decades the mTSP received great attention, and various approaches have been proposed to solve the problem. For this we may refer branch-and-cut (Borne et al. 2013), Cuckoo search (Ouaarab et al. 2013), Firefly algorithm (Ma et al. 2014), and Neural network (Hsu et al. 1991). The main goal is to minimize the total travelling cost of the above problem that is often formulated as assignment based integer linear programming (Bektas 2006). A new approach of chromosome representation, the so-called two-part chromosome technique has been developed by Carter and Ragsdale (2006) which reduces the size of the search space by the elimination of redundant solutions. In this genetic algorithm-based method, they effectively reduced redundant solutions in the search space. An evolutionary programming method has been proposed by Kota and Jarmai (2015) to solve mTSP. The swarm intelligence technique-based ACO (Applegate et al. 2002; Dorigo and Colorni 1996; Dorigo and Gambardella 1997a, b) has been first applied to TSP which is based on the foraging strategies of ants. The basic idea underlying this ant-based algorithm is to use a positive feedback mechanism, based on an analogy with the pheromone-laying, pheromone-following behaviour of some species of ants and some other social insects. Recently, Bertazzi and Maggioni (2014) have proposed a modern solution methodology for the stochastic capacitated travelling salesmen location problem with recourse.

Fuzzy set was introduced first by Zadeh (1965). Fuzzy-rough set is slightly younger than rough sets (Pawlak 1982). Rough set has left an important mark on the way we represent and compute with imperfect information nowadays. In some real-life problems both fuzziness and roughness exist simultaneously. Fuzzy-rough set theory is a well-known tool for processing different types of inaccurate, unreliable, and ambiguous data. To deal with these situations, the use of fuzzy-rough variables is very effective to model the problem. Still, fuzzy-rough set is an important field of research in theory of computation. The fuzzy-rough set was introduced by Dubois and Prade (1987, 1990). Then some researchers have modelled different problems where fuzziness and roughness coexist (Deer et al. 2015; Wang and Hu 2015; Radzikowska and Kerre 2002; Wang 2003; Shen and Richard 2004; Liu et al. 2006; Xu and Zhao 2008)

In all the above literature it is implicitly assumed that travelling cost from one node to another is fixed, i.e., crisp in nature. However, travelling cost from one node to another usually depends on the conveyance used for travelling. Travelling cost is also varied depending on the availability of conveyance. To the best of our knowledge till now none of the researchers used fuzzy-rough travel cost for mTSP.

In this paper, a solid multiple TSP has been considered where salesman can utilize different conveyances to travel form one city to another. In our model, costs for travelling between the cities using different conveyances is considered as fuzzy-rough in nature. This paper shows the results of different experiments in fuzzy-rough environment. The document is organized as follows: Sect. 2 presents all the prerequisite mathematics. Section 3 describes problem definition. Section 4 describes the algorithm in detail. Section 5 presents the results of computational experiments made by us and analyses the results. Finally, the paper is concluded in Sect. 6.

2 Prerequisite mathematics

Rough space Let Λ be a nonempty set, k be an σ-algebra of subset of Λ, Δ be an element in k, and π be a nonnegative, real-valued, additive set function. Then $(\Lambda, \Delta, k, \pi)$ is called a rough space.

When we do not have enough information to determine the measure of π for a real-life problem, we use Laplace criterion which assumes that all elements in Λ are equally likely to occur. For this case, the measure π may be taken as the cardinality of set Λ. This criterion is used in all examples in this paper, for the sake of simplicity.

Rough variable Liu (2002): A rough variable $\check{\xi}$ is a measurable function from the rough space $(\Lambda, \Delta, k, \pi)$ to the set of real numbers. That is, for every Borel set B of $\Re$, we have

$$\{\lambda \in \Lambda |\ \check{\xi}(\lambda) \in B\} \in k.$$

The lower and upper approximations of the rough variable are defined as $\underline{\check{\xi}} = \{\check{\xi}(\lambda) \mid \lambda \in \Delta\}$ and $\overline{\check{\xi}} = \{\check{\xi}(\lambda) \mid \lambda \in \Lambda\}$, respectively.

Rough expectation Liu (2002): let $\check{X}$ be a rough variable. The expected value of the rough variable $\check{X}$ is denoted by $\mathrm{E}(\check{X})$ and defined by

$$E(\check{X}) = \int_0^{\infty} \mathrm{Tr}(\check{X} \geq r)\mathrm{d}r - \int_{-\infty}^{0} \mathrm{Tr}(\check{X} \leq r)\mathrm{d}r \tag{1}$$

provided that at least one of the two integrals is finite.

Lemma 1 *Let $\check{\xi} = ([a, b][c, d])$ is a rough variable, where $c > 0$; then the expected value of $\check{\xi}$ is $E(\check{\xi}) = \frac{1}{4}(a+b+c+d)$*

Proof Since $\check{\xi} = ([a, b][c, d])$ is a rough variable and r is a crisp number, then from the definition of trust measure we have

$$\mathrm{Tr}\{\check{\xi} \geq r\} = \begin{cases} 0 \quad \text{for } d \leq r \\ \frac{d-r}{2(d-c)} \quad \text{for } b \leq r \leq d \\ \frac{1}{2}(\frac{d-r}{d-c} + \frac{b-r}{b-a}) \quad \text{for } a \leq r \leq b \\ \frac{1}{2}(\frac{d-r}{d-c} + 1) \quad \text{for } c \leq r \leq a \\ 1 \quad \text{for } r \leq c \end{cases}$$

$$\mathrm{Tr}\{\check{\xi} \leq r\} = \begin{cases} 1 \quad \text{for } r \leq c \\ \frac{r-c}{2(d-c)} \quad \text{for } c \leq r \leq a \\ \frac{1}{2}(\frac{r-c}{d-c} + \frac{r-a}{b-a}) \quad \text{for } a \leq r \leq b \\ \frac{1}{2}(\frac{r-c}{d-c} + 1) \quad \text{for } b \leq r \leq d \\ 1 \quad \text{for } d \leq r \end{cases}$$

Thus, the expected value of $\check{\xi}$ is calculated using Eq. (1) as follows:

$$E(\check{\xi}) = \int_0^\infty \mathrm{Tr}(\check{\xi} \geq r)\mathrm{d}r - \int_{-\infty}^0 \mathrm{Tr}(\check{\xi} \leq r)\mathrm{d}r$$

$$= \int_0^c 1 \,\mathrm{d}r + \int_c^a \frac{1}{2}\left(\frac{d-r}{d-c} + 1\right)\mathrm{d}r + \int_a^b \frac{1}{2}\left(\frac{d-r}{d-c} + \frac{b-r}{b-a}\right)$$

$$\mathrm{d}r + \int_b^d \frac{d-r}{2(d-c)}\mathrm{d}r = \frac{1}{4}(a+b+c+d)$$

Fuzzy-rough variable Liu (2002): Fuzzy-rough variable is a measurable function from a rough space $(\Lambda, \Delta, k, \pi)$ to the set of fuzzy variables. More specifically, a fuzzy-rough variable is a rough variable taking fuzzy values.

Fuzzy-rough expectation Liu (2002): Suppose $\overset{\vee}{\tilde{X}}$ is a fuzzy-rough variable. The expected value of the fuzzy-rough variable $\overset{\vee}{\tilde{X}}$ is denoted by $E(\overset{\vee}{\tilde{X}})$ and is defined by

$$E(\overset{\vee}{\tilde{X}}) = \int_0^\infty \mathrm{Tr}(\lambda \in \Lambda \mid E[\overset{\vee}{\tilde{X}}(\lambda)] \geq r)\mathrm{d}r - \int_{-\infty}^0 \mathrm{Tr}(\lambda \in \Lambda \mid E[\overset{\vee}{\tilde{X}}(\lambda)] \leq r)\mathrm{d}r \tag{2}$$

provided that at least one of the two integrals is finite. □

Lemma 2 *Let* $\overset{\vee}{\check{\xi}} = (\check{\xi} - L, \check{\xi}, \check{\xi} + R)$ *be a fuzzy-rough variable, where* $\check{\xi} = ([a, b][c, d])$ *is a rough variable; then the expected value of* $\check{\xi}$ *is*

$$E(\overset{\vee}{\tilde{X}}) = \frac{1}{4}(a+b+c+d) + \frac{\sigma R - (1-\sigma)L}{2}.$$

Proof Since $\overset{\vee}{\check{\xi}} = (\check{\xi} - L, \check{\xi}, \check{\xi} + R)$, where $\check{\xi} = ([a, b][c, d])$ is a rough variable, using the expectation of fuzzy variable we get,

$$E(\overset{\vee}{\check{\xi}}) = E\left[\frac{1}{2}\{(1-\sigma)(\check{\xi} - L) + \check{\xi} + \sigma(\check{\xi} + R)\}\right] = E[\check{\xi} + \theta],$$

where $\theta = \frac{\sigma R - (1-\sigma)L}{2}$ and $0 \leq \sigma \leq 1$.

Now, using Lemma 1 we get,

$$E(\check{\xi} + \theta) = \frac{1}{4}(a+b+c+d) + \theta = \frac{1}{4}(a+b+c+d) + \frac{\sigma R - (1-\sigma)L}{2}.$$

Therefore,

$$E(\overset{\vee}{\check{\xi}}) = \frac{1}{4}(a+b+c+d) + \frac{\sigma R - (1-\sigma)L}{2}, \quad \text{where } 0 \leq \sigma \leq 1.$$

□

3 Problem definition and notation

In this section, at first the mTSP is described in crisp environment and then in the fuzzy-rough environment.

3.1 Single depot solid mTSP

Usually, multiple TSP is formulated by integer programming formulations. However, in case of integer programming some part (variables) of the formulation may not be integer. In the proposed problem, the travelling cost is considered as real number. The mTSP is more capable to model real-life applications than TSP, since it handles two or more salesmen. In our proposed problem of solid mTSP, we have assumed that there are different conveyance facilities to travel from any city to any other city and travel cost is different for different conveyance/vehicle.

Consider a complete directed graph $G = (V, A, H)$, where V is the set of n nodes (vertices), A is the set of arcs, $C = c_{ijk}$ is the cost (distance) matrix associated with each arc$(i, j, k) \in A$ using kth type conveyance, and H is the set of conveyances'/vehicles' type. The cost matrix C can be symmetric, asymmetric, or Euclidean. Let there be m salesmen and all salesmen start from depot city 1 and return to the same, depot city. Thus, the single depot solid mTSP consists of finding tours for m salesmen such that all start and end at the depot city, but the tours for each pair of salesmen are likely to be different, along with their conveyances. The maximum and minimum number of nodes visited by a salesman lies within a predetermined interval, and the goal is to minimize the overall cost of visiting all the nodes.

Each salesman starts from depot city and visits a set of cities exactly once using suitable conveyances available at the cities and returns to the depot city at minimum cost. Let us define x_{ijk} as a binary variable equal to 1, if arc(i, j, k) is in the optimal solution and 0, otherwise. For any salesman, u_i is the number of nodes visited on that salesman's path from the origin up to city i (i.e., the visit number of the ith city). Each salesman may visit at most W cities and at least K cities; thus, $1 \leq u_i \leq W$ for all $i \geq 2$. When $x_{i1} = 1$, then $K \leq u_i \leq W$.

Thus, the proposed problem can be written as follows:

$$\left.\begin{array}{l}
Z = \text{Minimize} \displaystyle\sum_{(i,j)\in A \text{ and } k\in H} c_{ijk}x_{ijk} \\
\text{subject to } \displaystyle\sum_{j=2}^{n} x_{1jk} = m, \\
\displaystyle\sum_{j=2}^{n} x_{j1k} = m, \\
\displaystyle\sum_{i=1}^{n} x_{ijk} = 1,\ j = 2, 3, \ldots, n,\ \text{and } k \in H \\
\displaystyle\sum_{j=1}^{n} x_{ijk} = 1,\ i = 2, 3, \ldots, n,\ \text{and } k \in H \\
u_i + (W-2)x_{1ik} - x_{i1k} \leq W - 1;\ i = 2, 3, \ldots, n,\ \text{and } k \in H \\
u_i + x_{1ik} + (2-K)x_{i1k} \geq 2,\ i = 2, 3, \ldots, n,\ \text{and } k \in H \\
x_{ijk} \in \{0, 1\},\ \text{such that } (i, j) \in A.
\end{array}\right\} \tag{3}$$

The above formulation ensures that $2 \leq K \leq \lfloor (n-1)/m \rfloor$ and $W \geq K$. For minimum and maximum number of nodes visited by each salesman is defined by constants 5 and 6, and $u_i = 1$ if and only if i is the first node in the tour for any salesman.

3.2 Single depot solid mTSP with fuzzy-rough travel cost

If the cost of travel parameter of an objective function described in Eq. (3) is fuzzy-rough in nature, this type of problem is called the single-objective optimization problem under fuzzy-rough environment. Then the above problem may be represented in the following way:

$$\left.\begin{array}{l}
Z = \text{Minimize} \displaystyle\sum_{(i,j)\in A \text{ and } k\in H} \overset{\vee}{\widetilde{c_{ijk}}}x_{ijk} \\
\text{subject to } \displaystyle\sum_{j=2}^{n} x_{1jk} = m, \\
\displaystyle\sum_{j=2}^{n} x_{j1k} = m, \\
\displaystyle\sum_{i=1}^{n} x_{ijk} = 1,\ j = 2, 3, \ldots, n,\ \text{and } k \in H \\
\displaystyle\sum_{j=1}^{n} x_{ijk} = 1,\ i = 2, 3, \ldots, n,\ \text{and } k \in H \\
u_i + (W-2)x_{1ik} - x_{i1k} \leq W - 1;\ i = 2, 3, \ldots, n,\ \text{and } k \in H \\
u_i + x_{1ik} + (2-K)x_{i1k} \geq 2,\ i = 2, 3, \ldots, n,\ \text{and } k \in H \\
x_{ijk} \in \{0, 1\},\ \text{such that } (i, j) \in A.
\end{array}\right\} \tag{4}$$

Here $\overset{\vee}{\widetilde{c_{ijk}}}$ $(\check{c} - L, \check{c}, \check{c} + R)$ is assumed to be a fuzzy-rough variable, where $\check{c}_{ijk} = ([c_{ijk1}, c_{ijk2}], [c_{ijk3}, c_{ijk4}])$ is a rough variable.

Following Lemma 2, the Eq. (4) can be re-written as follows:

$$\left.\begin{array}{l}
Z = \text{Minimize} \left[E\left(\displaystyle\sum_{(i,j)\in A,\ k\in H} \overset{\vee}{\widetilde{c_{ijk}}}x_{ijk} \right) \right] \\
= \text{Minimize} \displaystyle\sum_{(i,j)\in A,\ k\in H} \left(\frac{(c_{ijk1}+c_{ijk2}+c_{ijk3}+c_{ijk4})}{4} + \left(\frac{\sigma R - (1-\sigma)L}{2} \right) \right) \\
\text{subject to } \displaystyle\sum_{j=2}^{n} x_{1jk} = m, \\
\displaystyle\sum_{j=2}^{n} x_{j1k} = m, \\
\displaystyle\sum_{i=1}^{n} x_{ijk} = 1,\ j = 2, 3, \ldots, n,\ \text{and } k \in H \\
\displaystyle\sum_{j=1}^{n} x_{ijk} = 1,\ i = 2, 3, \ldots, n,\ \text{and } k \in H \\
u_i + (W-2)x_{1ik} - x_{i1k} \leq W - 1;\ i = 2, 3, \ldots, n,\ \text{and } k \in H \\
u_i + x_{1ik} + (2-K)x_{i1k} \geq 2;\ i = 2, 3, \ldots, n, \\
x_{ijk} \in \{0, 1\},\ \text{such that } (i, j) \in A.
\end{array}\right\} \tag{5}$$

Thus, Eq. (5) describes the formulation of single depot mTSP in fuzzy-rough environment. In the experimental section we have depicted the computational results.

4 Algorithm description

In this paper, we have proposed a new genetic ant colony optimization technique, to solve the solid multiple travelling salesman problem. Now we briefly review the concepts of ant systems as well as ant colony systems from Colorni et al. (1994), Dorigo and Gambardella (1997a), and Dorigo et al. (1996). Then we briefly review the concepts of genetic algorithms (Goldberg 1989; Gwiazda 2006; Holland 1975).

4.1 Ant colony optimization algorithm

Assume that there are n cities, u ants and the initial pheromone (τ_{ij}) on each edge of solid mTSP is set to as per the cost of travel from city to city. Usually, pheromone of city i to city j is set to $1/\text{COST}_{ij}$, where COST_{ij} is the cost of travel from city i to city j. Each ant starts at the first city and visits the other cities according to the probabilistic selection (Colorni et al. 1991) on the basis of pheromone. After the ants complete their routes, the system evaluates the length of the routes. Then, the system uses the pheromone update rule to update the pheromone information. The learning procedure is to update the pheromone information repeatedly. The main ideas of an ant system are reviewed as follows (Colorni et al. 1991):

1. *Transition rule:* While an ant is at city i, if selects the next city j from the unvisited cities by the salesman with a probability p_{ij} given by the formula (using the Roulette-Wheel selection method):

$$p_{ij} = \frac{\tau_{ij}^{\alpha}}{\sum\limits_{j \in \text{NODE}} \tau_{ij}^{\alpha}}$$

Here the value of α is 1.6.
2. *Pheromone update rule:* The pheromone update operation is performed after all the ants complete their travelling routes, shown as follows:

$$\tau(i, j) = (1 - \rho).\tau(i, j) + \sum_{t=1}^{m} \Delta\tau(i, j)$$

3. *Pheromone evaporation:* Due to evaporation, an ant chooses to follow the path with the most intense pheromone trail, which is the shortest path. The evaporation of pheromone is performed before pheromone update using following formula:

$$\tau_{ij} = (1 - \rho)\tau_{ij}$$

with $\rho \in [0, 1]$. The constant, ρ, specifies the rate at which pheromone evaporates, causing ants to forget previous decisions. Here, to solve the problem the value of ρ is taken as 0.2.

4.2 Genetic algorithm

In this subsection, we have briefly reviewed the main concepts of genetic algorithms (Holland 1975). Genetic algorithm (GA) is an evolution system, observed in nature. The system encodes the parameters of a solution into a chromosome, where the basic element of a chromosome is the gene. In this area, we cite the works by Gen and Cheng (1997), Goldbarg et al. (2008), and Gwiazda (2006). The GA may be described using the following steps:

1. *Selection:* The GA often uses the Tournament selection or the Roulette-Wheel selection (Goldbarg et al. 2008) to choose better chromosomes into the gene pool until the gene pool is full, and this procedure is called mating pool. The selection of chromosomes with probability P_i according to Roulette-Wheel system is as follows:

$$P_i = \frac{\text{fitness}(i)}{\sum_{k=0}^{N-1} \text{fitness}(k)}$$

where fitness(i) is the fitness value of the ith chromosome and N is the size of the gene pool. The chromosomes with higher fitness values have high chance for selection into the gene pool.
2. *Crossover:* Crossover operation is used in GA to spend the searching area through the recombination of chromosomes. The system selects two chromosomes as parents and then reproduces their offspring after the crossover operation. When the crossover operation is over, the system puts their offspring into the gene pool. There are different types of crossover operations, such as one-point crossover, two-point crossover, cut-and-splice, etc. (Gwiazda 2006).
3. *Mutation:* The mutation operation is performed after the crossover operation is over. The traditional mutation method is to randomly select a gene of some selected chromosome, and then randomly change any value of the chromosome. The purpose of the mutation operation is to prevent the chromosomes in the new population to fall into a local minimum. The mutation operation continues until every chromosome is taken into consideration.
4. *Evaluation:* After the crossover and the mutation, GA evaluates the fitness value of each chromosome. If the maximum number of generations is reached or the termination condition is met, the chromosome with the highest fitness value is considered as the best solution. Otherwise, the process, is repeated a constant/resonable number of times.

4.3 The proposed genetic ant colony optimization algorithm

In this paper we have proposed a new Genetic Ant Colony based algorithm to solve the proposed problem. At first we present the pseudo-code for the proposed genetic ant colony optimization algorithm and then the algorithm is described in detail for the proposed problem.

Genetic ant colony optimization algorithm( )

(a) Initialization of population:
For 1 to *popsize* repeat the following steps. [*popsize* is the size of the population in GA.]
(i) Randomly generate a set of N random numbers between 1 and N without repetition and each number is again associated with a vehicle type.
(ii) Divide the sets into m subparts/subgroups (for m salesmen problem) except the depot city.
(iii) Rearrange each subgroup/subtour using ACO algorithm (call subroutine *ALGO_ACO*( )).
End For
(b) Evaluation:
Do While (the termination condition has not reached)

(i) Selection for mating pool
(ii) Cyclic crossover
(iii) Mutation
(iv) Sequentially extract or sub-divide each chromosome into m subparts
for each salesman path
(v) Rearrange each subtour using ACO algorithm using
subroutine $ALGO\_ACO$
(v) Calculate fitness for each chromosome of population
End While
(c) Extract optimal solution from GA
(d) Stop

Subroutine *ALGO_ACO*( )
(a) Initialize pheromone for each subtour on the basis of travelling cost from city to city
(b) Evaluation:
Do While (the termination condition has reached or not converged)
(i) Construct path/solution for each ant
(ii) Pheromone evaporation
(iii) Pheromone update
(iv) Refinement
End While
(c) Extract the best path and return

Description of genetic ant colony optimization algorithm: Ant colony based algorithms are becoming popular approaches for solving combinatorial optimization problems in literature. These had been introduced by Dorigo and Gambardella (1997a). The fundamental concept of ant heuristics is based on the behaviour of natural ants that succeed in finding the shortest paths from their nest to food sources by communicating via a collective memory that consists of pheromone trails. However, it tends to follow a path with a high pheromone level when many ants move in a common area, which leads to an autocatalytic process. Finally, the ant does not choose its direction based on the level of pheromone exclusively, but also takes the nearness of the nest and of the food source into account. This allows the ants to find shorter paths.

In the proposed genetic ant based algorithm, each chromosome (or complete tour) is randomly generated and then subdivided into same number of parts (or subtours) as the number of salesmen. Then each subtour is arranged by the ACO algorithm. The used ACO for a subtour construction is again modified (on simple ACO) by the refinement operation. In the refinement operation, after each generation the worst ant path is replaced by the best ant path, i.e., the worst ant must choose the path of the ant which gives the best solution by force. After this as usual the mating pool is performed. Then, crossover and mutation, two genetic operations are carried out one after the other. This procedure is continued until the stopping criteria is met.

Solution construction: At first, we randomly generate a complete solution like: $\{1, 5, 3, 4, 7, 9, 6, 8, 2, 10\}$, if 1 is the depot city; then the rest of the cities are sub-divided into m parts, where m is the number of salesmen. Let us assume that the subsets are $\{5, 3, 4, 7\}$, $\{9, 6\}$, and $\{8, 2, 10\}$. Now for each city a conveyance type is randomly generated and presented as a number pair of city and vehicle type, i.e., a set of number pairs is generated for each salesman. A number pair (3, 2) signifies that the traveller used type 2 conveyance/vehicle to travel from city number 3. We assume, for our example, $\{(5, 3), (3, 2), (4, 1), (7, 1)\}$, $\{(9, 2), (6, 1)\}$, $\{(8, 3), (2, 2), (10, 1)\}$, and $\{(1, 3)\}$ as a solution (this implies that from the depot city all salesmen use only type 3 conveyance). For the depot city, a conveyance type is randomly generated. After this, we rearrange each subpart by the ACO algorithm, which is based on the pheromone or trail information that is used to assign on the basis of travel cost. For ACO, the trail information is assigned on the basis of travel cost, i.e., for high travel cost the value of pheromone is low and for low travel cost the value of pheromone is high. The pheromone from ith city to jth city using kth type conveyance is $1/C_{ijk}$, where C_{ijk} is the cost of travel from ith city to $j$thcity using kth vehicle. After applying ACO to each subtour the rearranged subtours are $\{(3, 2), (5, 3), (7, 1), (4, 1)\}$, $\{(6, 1), (9, 2)\}$, and $\{(2, 2), (8, 3), (10, 1)\}$, respectively. Then exact tours of the salesman become $\{(1, 3), (3, 2), (5, 3), (7, 1), (4, 1)\}$, $\{(1, 3), (6, 1), (9, 2)\}$, and $\{(1, 3), (2, 2), (8, 3), (10, 1)\}$, respectively, because all salesmen must return back to the depot city. One chromosome is described pictorially in Fig. 1.

4.4 Genetic ant colony optimization algorithm for solid mTSP

(a) Representation: If the number of cities is N, population size is P, and the number of salesmen is M, then we consider a four dimensional integer array POPU$[P][M][N][V]$ to represent path as well as used conveyance type of each individual. For construction of different salesmen's tour (subtours) with corresponding vehicle by using ACO, we use a three dimension array PATH$[M][N][V]$ to hold the subtour (with corresponding vehicle type). PATH$[i][j][k]$ represents path of ith salesman. PATH$[i][j][k]$ represents conveyance used by ith salesman to travel from jth node (PATH$[k][i][k]$) to $(j + 1)$th node PATH($[i][j + 1]$) using kth vehicle. Let us assume that the number of ants be N_{ant}.

(b) Initialization: pop_size number of such solutions $X_i = (x_{i1}, x_{i2}, \ldots, x_{iN})$, $i = 1, 2, \ldots,$ pop_size, are randomly generated by random number generator, such that each solution satisfies the constraints of the problem. This complete solution X_i (of a tour)is subdivided into M subtours and

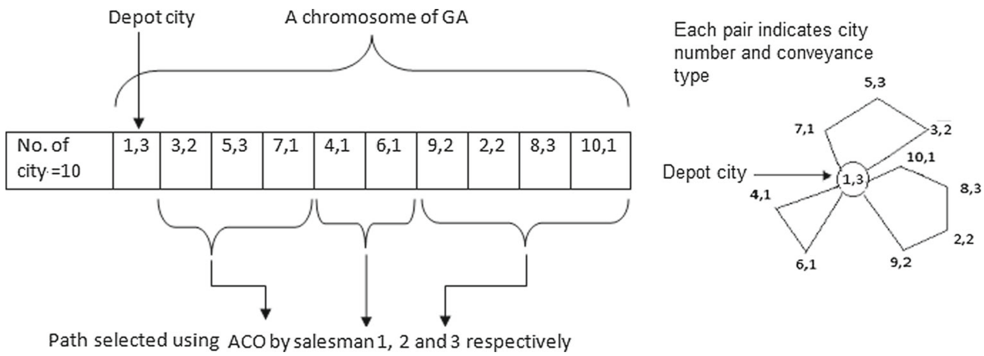

Fig. 1 Representation of one chromosome using three salesmen

each subtour is to be visited by one salesman only. X_i is divided into M subtours of cities $S_1, S_2, \ldots, S_M$. Again, each S_t $(t = 1, 2, \ldots, M)$ is some set of cities like S_{t1}, S_{t2}, and so on, such that for different values of t the sets are mutually disjoint to each other. Here, the shortest (or minimum cost) subtour is to be chosen by ACO algorithm.

(c) Evaluation process: To find minimum cost, for a certain combination of subtours we use ACO, that is for S_1, as an example, S_{11}, S_{12}, and so on are so chosen such that it needs minimum cost. This procedure is allowed to all remaining S_t's, $2 \leq t \leq M$. For each such S_t $(t = 1, 2, \ldots, M)$, we execute the steps below to find the respective shortest path.

ACO procedure for subtour used by any individual salesman:

(i) *Pheromone initialization:* As the aim of a TSP is to minimize the cost, it is assumed that the initial value of TAO$[i][j[k]$ is τ_{ijk} and the value of τ_{ijk} $= (1/\text{COST}[i][j][k])$, where COST$[i][j[k]$ is the cost of travel by ith salesman from jth city using kth vehicle.

(ii) *Path selection:* Let the recent position of ant k be node i using vehicle v. Then the next node $j \in S_t$ (for tth subtour) is to be selected by the traveller using conveyances v with a probability p_{ijv}. The value of p_{ijv} is calculated by the following formula:

$$p_{ijv} = \frac{T_{ijv}^{\alpha}}{\sum_{j \in S_t} \sum_{v=1}^{M} T_{ijv}^{\alpha}},$$

where α is a positive constant as described above. In this case, the Roulette-Wheel selection process (Michalewicz 1992) is used. Here, to solve the problem, the value of α is taken as 1.6.

(iii) *Pheromone evaporation:* For evaporation of pheromone the following formula is used

$$\tau_{ijk} = (1 - \rho)\tau_{ijk}$$

with $\rho \in [0, 1]$. The constant, ρ specifies the rate at which pheromone evaporates, causing ants to forget previous decisions. Here, to solve the problem, the value of ρ is taken as 0.2.

(iv) *Pheromone updating:* Once all ants have constructed their complete tour, pheromone is increased on the paths through which the ants move. If for solid mTSP, COST(k) be the cost of PATH$[k]$ (for kth ant), then the pheromone of $\tau_{\text{PATH}[k][i]\text{PATH}[k][i+1]V[k][i]}$ is increased by $1/\text{COST}(k)$ (for each pair of cities).

(v) *Refinement:* After each iteration, if all ants do not select same path or not give the same solution, then a worst path of ant is replaced by a best path of ant, i.e., a worst solution is replaced by a best solution.

(vi) *Stopping criteria:* When all or at least 80 % of the ants follow the same path, then this path is considered as a solution.

With the help of the above steps, different salesmen find subtours and then combine the paths of different salesmen to construct a chromosome.

To find fitness of a solution (or complete subtour) X_i $[(X_i, V_i)$ for solid mTSP], the following two steps are performed one after another.

- Calculate objective function value OBJ$_i$ for the solution X_i $[(X_i, V_i)$ for solid mTSP].
- As the problems are of minimization type and take MVAL $-$ OBJ$_i$ as fitness, FIT$_i$ of X_i $[(X_i, V_i)$ for solid mTSP], where MVAL is a sufficiently large value to make the fitness positive.

Roulette-Wheel selection process for mating pool: The following steps are followed for selection of $p(t)$ from $p(t-1)$ (Michalewicz 1992):

(i) To obtain the total fitness of population, $p(t-1)$, we use the formula: $F_{\text{tot}} = \sum_{j=1}^{\text{pop\_size}} \text{FIT}_j$.

(ii) Calculate the probability of selection p_i of each solution X_i $[(X_i, V_i)$ for solid mTSP] by the formula $p_i = \frac{\text{FIT}_i}{F_{\text{tot}}}$.

(iii) Calculate the cumulative probability q_i for each solution X_i $[(X_i, V_i)$ for solid mTSP] by the formula $q_i = \sum_{k=1}^{i} p_k$.

(iv) A random number r is generated from the range [0, 1].
(v) If $r < q_1$, then select X_1 [(X_1, V_1) for solid mTSP]; otherwise, select X_j $(2 \leq j \leq n)$ [(X_j, V_j) for solid mTSP], where $q_{j-1} \leq r < q_j$.
(vi) Repeat Steps (iv) and (v) pop_size times to select pop_size solutions for mating pool. Clearly one solution may be selected more than once.
(vii) Selected solution set is denoted by $p(t)$ in the proposed GA algorithm.

Crossover:

(i) *Selection for crossover:* For each solution of $p(t)$, generate a random number r from the range [0, 1]. If $r < p_c$, then the solution is taken for crossover.
(ii) *Crossover process:* For solid mTSP cyclic crossover process (Pratihar 2008) is employed. The cyclic crossover focuses on subsets of cities that occupy the same subset of positions in both parents. Then, these cities are copied from the first parent to the offspring (at the same positions), and the remaining positions are filled with the cities of the second parent. In this way, the position of each city is inherited from one of the two parents. However, many edges can be broken in the process, because the initial subset of cities is not necessarily located at consecutive positions in the parent tours.

Mutation:

(i) *Selection for mutation:* For each solution of $p(t)$, generate a random number r from the range [0, 1]. If $r < p_m$, then the solution is taken for mutation.
(ii) Mutation process: For solid multiple TSP to mutate a solution (X, V), where $X = (x_1, x_2, \ldots, x_N)$ and $V = (v_1, v_2, \ldots, v_N)$, at first an integer is randomly selected in the range [1, 2]. If 1 is selected, then another two random integers i and j are selected in the range $[1, N]$. Then interchange x_i and x_j to get a child solution. If 2 is selected, then another two random integers i and j are selected in the range $[1, N]$. Values of v_i and v_j are interchanged to get another child solution. If a child solution satisfies the constraint of the problem, then it replaces the parent solution.

Computing configuration: All computational experiments are conducted with Dev C++ 4.9.9.2, Core i3 Processor, Windows 7 Operating System and 2 GB RAM.

5 Numerical illustration

In this section, we have analysed the performance of the algorithm with different data set. For this at first we consider some benchmark instances from TSPLIB and compare the results with existing literature.

5.1 Performance analysis using TSPLIB data set

In order to test the performance of the proposed genetic ant colony optimization based algorithm, we have considered some benchmark data sets, taken from TSPLIB. Table 1 presents a comparison of the experimental results of the proposed method with an existing method (Tsai 2006). In this section, we have compared the computed results with respect to the best solutions for 20 independent runs. Here, we compare the results of multiple travelling salesman problem considering different numbers of salesmen (2, 3, and 4 salesmen). For comparison we have considered only travel cost of each instance with same number of salesmen. For these instances we set number of ants as 30. Maximum number of generations for GA is set to 1000 for all these instances. Percentage of crossover and mutation is considered as 60. Here, the value of α and ρ are taken as 1.6 and 0.2, respectively.

In ACO, α and ρ are adjustable positive parameters that control the relative weights of the pheromone trail and make the heuristic visible. Changdar et al. (2013) have shown that in case of pheromone update the ACO algorithm gives significantly better results for $\alpha = 1.6$. Changdar et al. (2013) have also stated that the ACO algorithm provides better results when the value of ρ is closed to 0.2. Furthermore, Engelbrech (2005) has described that the value of ρ varies from 0 to 1 and the value of α must be positive. For our model, we have set the values of α and ρ to 1.6 and 0.2, respectively. Moreover, during the test runs it has been observed that further changing the values of α and ρ. from the said values the results deviate from the desired ones.

The blank results for *ftv35* and *ftv38* indicate that the solutions are not given in Tsai 2006. From Table 1, we observe that our proposed algorithm computes better result or same result except *ftv33* (considering three salesmen) and *ftv35* (considering two salesmen). These results do show that our proposed approach is computationally significant, efficient and acceptable.

To visually compare the performance of the proposed genetic ant colony optimization algorithm with other two optimization algorithms, four benchmark instances are shown in Figs. 2, 3, 4, and 5. We have plotted four bar charts to compare the travel cost considering different number of salesmen for *br17*, *ftv33*, *ftv35*, and *ftv38*, respectively. We have plotted the graphs for the instances *br17* and *ftv33* considering two, three, and four salesmen. We have plotted the graphs for the instances *ftv55* and *ftv38* considering two and three salesmen.

Figures 2, 3, 4 and 5 help to synthesize the results of four benchmark instances *br17, ftv33, ftv35*, and *ftv38*, respec-

Table 1 Results for comparison of multiple asymmetric TSP using different algorithms

Problem name	Number of salesmen	Best integer by CPLEX	Best integer by Benders method	Best integer by the proposed algorithm
	2	39	39	39
br17	3	42	42	42
	4	47	47	47
	2	1302	1302	1302
ftv33	3	1328	1328	1342
	4	1367	1367	1352
	2	1489	1489	1511
ftv35	3	1541	1511	1511
	4	–	1551	1532
	2	1551	1505	1505
ftv38	3	1567	1521	1521
	4	–	1546	1532

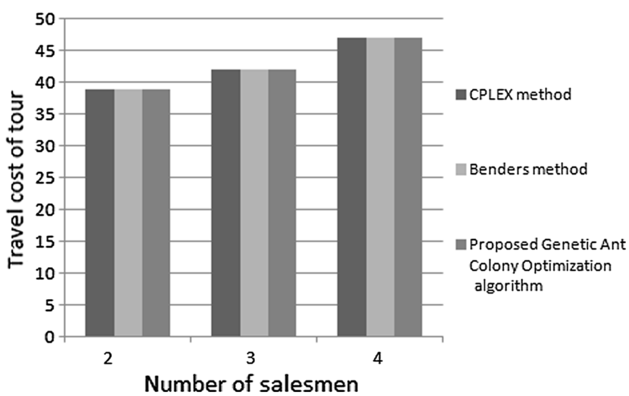

Fig. 2 Comparative travel cost for instance *br17* considering two, three, and four salesmen by bar chart

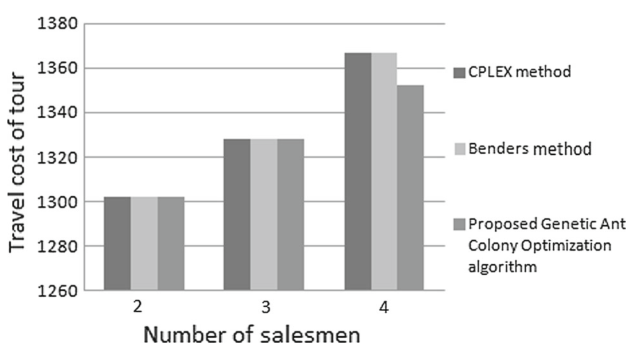

Fig. 3 Comparative travel cost for instance *ftv33* considering two, three, and four salesmen by bar chart

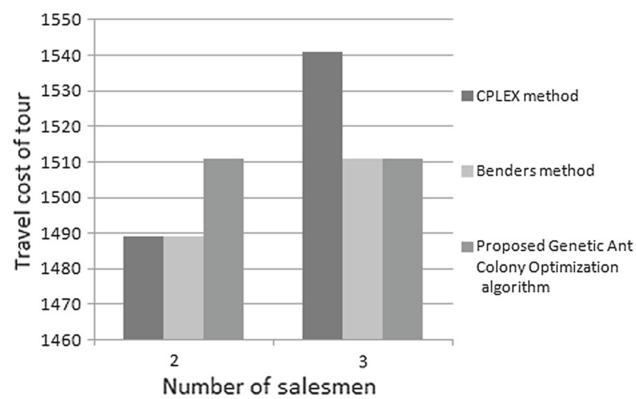

Fig. 4 Comparative travel cost for instance *ftv35* considering two and three salesmen by bar chart

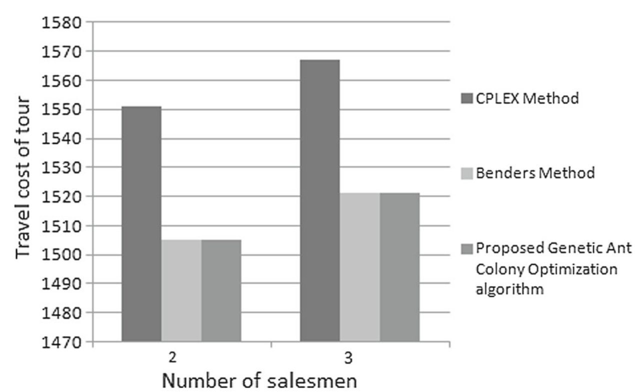

Fig. 5 Comparative travel cost for instance *ftv38* considering two and three salesman by bar chart

tively. They show the optimum results found by the proposed algorithm and other two existing algorithms, namely CPLEX method and Benders method, considering different number of salesmen. From the figures we have observed that in most of the cases our proposed MOGA computes better results.

5.2 Computational results using randomly generated crisp data set

In this section, we have presented the results of solid mTSP using randomly generated cost for 100, 200, and 300 cities.

Table 2 Computational results of solid mTSP using randomly generated data sets for instances of 50-, 100-, and 200-city problem

No. of salesmen	50-city problem		100-city problem		200-city problem	
	Parameters	Cost	Parameters	Cost	Parameters	Cost
5	Min city: 5	1105.56	Min city: 10	2250.62	Min city: 15	4607.37
	Max city: 20		Max city: 30		Max city: 60	
	NOV: 4		NOV: 4		NOV: 4	
6	Min city: 5	1111.32	Min city: 5	2258.32	Max city: 15	4626.94
	Min city: 15		Max city: 30		Max city: 60	
	NOV: 4		NOV: 4		NOV: 4	
7	Min city: 2	1115.07	Min city: 5	2289.48	Min city: 5	4656.34
	Max city: 15		Max city: 30		Max city: 30	
	NOV: 4		NOV: 4		NOV: 4	

Table 3 Computational results of solid mTSP for given real data set in Table 5 (in "Appendix")

No. of salesmen	σ	Cost (objective value)	Path
2	0.0	148.91	First salesman (3) : : [1 : 1][4 : 2][5 : 1][2 : 1]
			Second salesman (3) : : [1 : 1][7 : 1][3 : 1][6 : 1]
	0.5	154.26	First salesman (4) : : [1 : 1][5 : 1][3 : 2][7 : 1][4 : 1]
			Second salesman (2) : : [1 : 1][2 : 1][6 : 1]
	1.0	159.45	First salesman (4) : : [1 : 2][2 : 1][5 : 1][3 : 1][4 : 1]
			Second salesman (2) : : [1 : 2][7 : 1][6 : 1]

To generate data sets we use rand() function of C programming language. For the experiment of this subsection, both percentages of crossover and mutation are assumed as 60. Here we consider either three or four types of conveyance facilities in solid mTSP. Types of vehicle are for different experiments mentioned in the result table. To generate the cost of first, second, and third type conveyance, random numbers are generated within the range [16.1, 18.2], [13.5, 15.8], and [12.8, 13.7], respectively. All random numbers are generated using seed = 1 of C language. The results of solid mTSP for 50, 100, and 200 cities problem instances are presented in Table 2. Here we include average results among 25 runs of the associated program on different seed for each problem. We set population size to 100, 220, and 320 for the problem with number of cities of 100, 200, and 300, respectively. Here presentation is shown in the format of $(u : v)$ means from uth city the salesman uses vth type vehicle. For the proposed genetic ant colony optimization algorithm, we set the maximum number of iterations as 2000, the value of α as 1.6, and the value of ρ as 0.2 . In each of our considered problems all cities have been covered/visited by different salesmen except the depot city (here we assume it the first city), such that the total city covered by different salesmen is one less than the total number of cities. The number of ants is assumed as 50 for these experiments.

Here in Table 2, Max city indicates the maximum number of cities a salesman is allowed to visit and Min city indicates the minimum umber of cities that must be visited by a salesman. NOV is the number of available vehicle types. In this table, we observe that the trend of results is quite correct as the cost of travel is changing according to the number of salesmen is changed. We again observe that the travel cost of the instance of 100-city problem is almost twice of the 50-city problem instance and that for 200-city problem instance is almost twice of the 100-city problem instance.

5.3 Computational results using a small-size real dataset in fuzzy-rough environment

In this section, we have presented the results of a 7-city solid mTSP in fuzzy-rough environment. Fuzzy-rough cost matrix is given in appendix (cf. "Appendix" Table 5). For this problem, we consider only two salesmen. In this case, the minimum and the maximum number of visited cities are 2 and 4, respectively. We set the population size 50 for this experiment. For these experiments, the number of ants is assumed as 20. For the proposed algorithm, the maximum number of iterations is considered as 1000, the value of α as 1.6 and ρ as 0.2. For this experiment as well, 'city number 1' is the depot city. For this experiment we have presented the

Table 4 Computational results (using random data set) for 50-, 100-, and 200-city problem in fuzzy-rough environment

No. of salesmen	50-city problem		100-city problem		200-city problem	
	σ	Cost	σ	Cost	σ	Cost
	0.0	1120.99	0.0	2414.74	0.0	4852.75
	0.2	1155.23	0.2	2450.44	0.2	4917.59
Five salesmen	0.5	1181.61	0.5	2494.69	0.5	5000.70
	0.8	1198.55	0.8	2534.08	0.8	5083.00
	1.0	1218.45	1.0	2570.42	1.0	5151.43
	0.0	1115.15	0.0	2432.27	0.0	4918.76
	0.2	1139.05	0.2	2461.43	0.2	4981.25
Six salesmen	0.5	1153.01	0.5	2505.24	0.5	5066.29
	0.8	1167.82	0.8	2576.36	0.8	5154.92
	1.0	1185.16	1.0	2591.73	1.0	5214.74
	0.0	1133.8	0.0	2457.65	0.0	4939.72
	0.2	1161.48	0.2	2496.98	0.2	5000.91
Seven salesmen	0.5	1186.81	0.5	2520.43	0.5	5094.53
	0.8	1218.42	0.8	2547.12	0.8	5179.51
	1.0	1221.54	1.0	2581.64	1.0	5237.54

best result after 15 runs of proposed algorithm in different seeds of C programming language. Here results is shown in the format of (u: v) means from uth city the salesman uses vth type vehicle.

From Table 3, we observe that when the value of σ is increased then the objective value is also increased. This fact is conceptually correct to the fuzzy-rough formulation of the objective function.

5.4 Computational results using randomly generated data set in fuzzy-rough environment

In this section, we have presented the experimental results for a large number of randomly generated instances of the problem under consideration through implementation of the algorithm developed in this paper in fuzzy-rough environment. To generate rough cost for each ith city to jth city, at first we generate three random numbers x, y, and z within the range [20.2, 25.0], [26.0, 30.0], and [31.0, 36.0] for first, second, and third type conveyance, respectively. If $([a, b][c, d])$ is rough cost ($c \leq a \leq b \leq d$), then to obtain the rough cost of each type conveyance two random numbers ξ_1 and ξ_2 are generated within the range [3.0, 4.0] and ξ_1 is subtracted to get the second component 'a' and ξ_2 is added to get the third component 'b' of rough cost. Another set of two random numbers ξ_3 and ξ_4 are generated within the range (2.0, 3.0), and then ξ_3 is subtracted from 'a' to obtain the first component 'c' and ξ_4 is added to obtain the last component 'd' of each cost. To convert each rough number into fuzzy-rough number, two random numbers L and R (as described in Sect. 2) are randomly generated within the range [1.0, 2.0]. For these experiments (to generate input data sets) all random numbers are generated in seed 1 of C language. We have presented the results for 50-, 100-, and 200-city problem instances considering five, six, and, seven salesmen in Table 4, respectively. In all the cases we have computed the cost (objective value) for different values of σ used in fuzzy-rough formulation of the proposed problem. For 50-city problem, the minimum number of cities visited by a salesman is fixed as 6 and the maximum number of cities he may be allowed to visit is 15; for 100-city problem, these parameters are 6 and 20, and for 100 cities, these parameters are 12 and 30, respectively. The values of ρ and α are considered as 0.2 and 1.6, respectively, for this experiment. The population size is set to 100 for both 50- and 100-city problem instance, and furthermore 250 for 200-city problem instance. The number of ants is considered as 50 for all the experiment of this section. The percentage of crossover and mutation is assumed as 55 and 45, respectively for these experiments. Maximum number of iteration is set to 5000 for 50- and 100-city problem and 1000 for 200-city problem. We have presented the best result after 15 runs of proposed algorithm in different seeds of C programming language.

In Table 4, we have observed that the trend of results is similar to the previous experiments, i.e., when the value of σ is increased, then the travel cost of tour is also increased. This trend is expected in fuzzy-rough environment. Thus, we conclude that our proposed algorithm works properly and generates desired results.

Now we have presented a graphical representation for sensitivity analysis of fuzzy-rough formulation. Using the fuzzy-rough experiments of 50-city and 200-city problem instances (as described in this section), we have explained

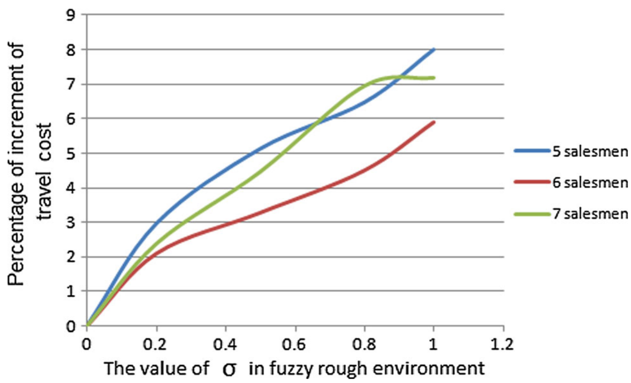

Fig. 6 Graph for percentage of increment of travel cost with respect to the values of σ for 50 city problem considering five, six, and seven salesmen

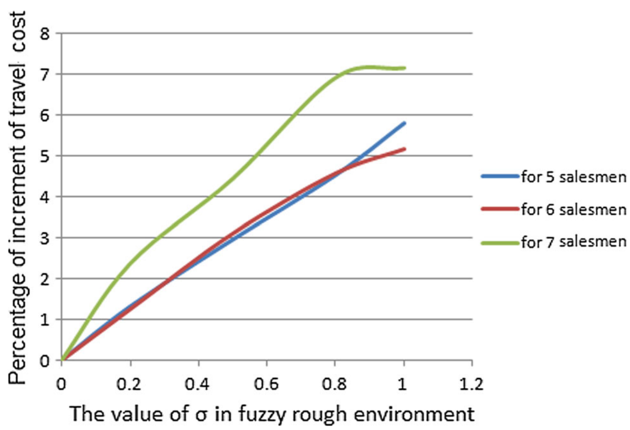

Fig. 7 Graph for percentage of increment of travel cost with respect to the value of σ for 200 city problem considering five, six, and seven salesmen

the graphical representation. For this we have plotted the graph for different values of σ versus objective values, i.e., travel costs for 50-city and 200-city problems in Figs. 6 and 7, respectively. For each problem we consider five, six, and seven salesmen and present accordingly.

We have obtained that the percentage of change of optimal travel cost (by the proposed algorithm) is lying between 4 and 8 (approximately) for both the instances of 50-city and 200-city problem. We again observed that for five, six, and seven salesmen the percentage of change of optimal cost remains in the same range [4, 8]. Thus, the optimal cost varies for different values of σ. From the fuzzy formulation given in Lemma 2 of Sect. 2, we observe that the value of parameter σ changes the objective value of the problem under consideration. The objective value (i.e., the travel cost of tour) for row $= 0$ is greater than that for row $= 1$. Hence, the trend of obtained results is quite pragmatic with the fuzzy-rough formulation.

6 Conclusion

In this paper, we have proposed a new genetic ant colony optimization based algorithm for multiple TSP. Here, we actually solve solid mTSP, where a traveller may use different conveyance facility to travel from one city to another city. The proposed algorithm has been developed with some properties of GA and some properties of ACO. To show the effectiveness of the algorithm we use some benchmark instances and compare the results with other existing algorithms. From the obtained results, it can be concluded that the newly added features enhance the performance efficiency of the algorithm. The trend of the results of fuzzy rough data sets is realistic and appropriate with fuzzy rough formulations.

In some of our future works, we would like to focus on the area of multi-objective multiple travelling salesman problems with fuzzy random travel cost. Also our future plan is to develop a soft computing technique using simulated annealing and ant colony optimization, which can solve multiple TSP with the help of fuzzy random number. This would also try to increase the solution quality in fuzzy environment.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Appendix

In Table 5, each fuzzy-rough travel cost has six components; the first and the last components are L and R, respectively as described in Lemma 2. The second, third, fourth, and fifth components are a, b, c, and d, respectively (rough number components), that have also been described in Lemma 2. Here 1stvh, 2ndvh, and 3rdvh indicate cost of travel using the first, second, and third type vehicle, respectively.

Table 5 Fuzzy-rough cost matrix for 7-city problem

	City / City	1	2	3	4
1	1st vh	16.93, 23.73, 14.43, 25.90, 1.72, 1.48	19.20, 26.25, 16.91, 28.63, 1.42, 1.72	18.41, 25.42, 15.75, 27.56, 1.71, 1.25	20.95, 27.35, 18.91, 30.29, 1.26, 1.65
	2nd vh	23.47, 30.90, 20.76, 33.04, 1.28, 1.83	24.98, 32.32, 22.25, 35.09, 1.54, 1.87	25.92, 33.12, 23.26, 35.87, 1.04, 1.86	23.98, 31.67, 21.25, 34.04, 1.35, 1.01
	3rd vh	32.31, 39.80, 29.37, 42.63, 1.44, 1.39	31.89, 38.86, 29.86, 41.75, 1.70, 1.81	30.87, 38.14, 28.10, 40.46, 1.03, 1.19	28.11, 35.05, 25.48, 37.68, 1.08, 1.95
2	1st vh	20.62, 27.43, 18.20, 30.37, 1.51, 1.03	19.74, 26.38, 16.90, 28.75, 1.02, 1.60	18.71, 26.33, 16.07, 28.82, 1.81, 1.42	18.27, 25.12, 15.66, 28.11, 1.70, 1.20
	2nd vh	24.48, 31.55, 21.89, 34.31, 1.65, 1.41	22.74, 29.28, 20.07, 31.77, 1.28, 1.73	23.62, 31.05, 21.11, 33.36, 1.62, 1.93	24.85, 31.28, 22.32, 33.87, 1.31, 1.50
	3rd vh	29.17, 36.33, 26.62, 38.81, 1.60, 1.04	27.27, 34.68, 24.33, 37.58, 1.79, 1.13	29.66, 36.50, 27.14, 39.06, 1.80, 1.30	30.04, 36.70, 27.42, 39.28, 1.80, 1.80
3	1st vh	20.26, 27.36, 18.11, 30.08, 1.19, 1.98	18.95, 26.23, 16.26, 28.76, 1.55, 1.44	18.05, 25.50, 15.78, 27.82, 1.36, 1.18	18.14, 24.43, 15.60, 26.88, 1.17, 1.47
	2nd vh	23.95, 31.01, 21.52, 33.56, 1.43, 1.55	25.52, 32.66, 23.32, 34.95, 1.42, 1.29	22.30, 30.02, 19.47, 32.97, 1.31, 1.76	22.52, 29.47, 20.08, 31.72, 1.02, 1.15
	3rd vh	29.69, 36.35, 27.63, 39.07, 1.75, 1.14	28.04, 34.81, 25.58, 37.54, 1.11, 1.70	29.05, 36.25, 26.06, 38.73, 1.14, 1.20	29.81, 37.20, 27.62, 39.51, 1.47, 1.02
4	1st vh	19.95, 27.51, 17.67, 30.06, 1.14, 1.69	16.82, 23.41, 14.67, 26.21, 1.09, 1.06	18.96, 26.01, 16.34, 28.64, 1.63, 1.93	17.02, 23.35, 14.25, 25.77, 1.55, 1.15
	2nd vh	25.16, 31.80, 23.03, 34.38, 1.69, 1.66	26.69, 33.46, 24.52, 35.78, 1.58, 1.23	24.17, 31.59, 21.21, 33.72, 1.60, 1.74	24.29, 32.00, 22.10, 34.77, 1.94, 1.85
	3rd vh	29.14, 35.98, 26.46, 38.57, 1.85, 1.48	27.46, 34.38, 25.35, 37.26, 1.09, 1.57	29.11, 35.83, 26.85, 38.09, 1.20, 1.12	30.48, 38.21, 27.77, 41.17, 1.58, 1.37
5	1st vh	18.79, 26.36, 16.31, 29.27, 1.63, 1.07	21.48, 28.47, 18.63, 30.86, 1.28, 1.70	20.45, 27.45, 17.99, 30.06, 1.88, 1.42	20.37, 27.37, 18.33, 29.85, 1.68, 1.09
	2nd vh	25.72, 33.33, 23.49, 35.47, 1.75, 1.51	23.04, 30.41, 20.50, 32.97, 1.99, 1.18	24.96, 32.15, 22.66, 34.83, 1.04, 1.31	24.40, 31.98, 22.25, 34.33, 1.31, 1.65
	3rd vh	31.53, 38.16, 28.71, 40.39, 1.87, 1.63	30.56, 38.07, 27.68, 40.37, 1.41, 1.64	32.30, 39.19, 29.69, 41.21, 1.93, 1.11	31.66, 37.96, 29.32, 40.72, 1.19, 1.61
6	1st vh	17.80, 25.03, 14.83, 27.90, 1.91, 1.08	17.86, 25.33, 15.53, 27.46, 1.47, 1.15	16.68, 24.26, 14.27, 26.72, 1.05, 1.68	20.72, 27.33, 17.94, 29.36, 1.23, 1.18
	2nd vh	24.47, 31.73, 21.79, 34.49, 1.26, 1.52	22.78, 30.11, 19.88, 32.21, 1.42, 1.53	25.01, 31.96, 22.95, 34.18, 1.76, 1.95	22.30, 29.78, 20.08, 32.53, 1.92, 1.36
	3rd vh	32.64, 39.87, 29.77, 42.41, 1.25, 1.51	30.03, 37.01, 27.47, 39.04, 1.54, 1.35	32.29, 39.13, 29.34, 41.97, 1.43, 1.62	28.52, 36.23, 25.56, 38.55, 1.73, 1.41
7	1st vh	18.51, 24.83, 16.48, 27.03, 1.08, 1.32	19.66, 26.55, 17.26, 29.48, 1.83, 1.37	19.73, 26.70, 17.32, 29.16, 1.23, 1.51	20.78, 27.15, 18.57, 29.44, 1.08, 1.95
	2nd vh	26.38, 32.59, 24.02, 34.85, 1.90, 1.68	26.25, 32.52, 23.63, 35.49, 1.00, 1.95	25.89, 33.34, 23.37, 35.61, 1.94, 1.87	22.63, 29.82, 19.86, 32.20, 1.82, 1.95
	3rd vh	30.31, 38.12, 27.74, 40.61, 1.95, 1.28	31.40, 38.30, 28.72, 40.61, 1.70, 1.59	27.59, 35.26, 25.33, 37.38, 1.22, 1.55	30.59, 37.38, 28.48, 40.28, 1.07, 1.79

Table 5 continued

	City / City	5	6	7
1	1st vh	19.99, 27.79, 17.61, 30.73, 1.31, 1.94	20.69, 27.97, 18.01, 30.36, 1.02, 1.75	19.77, 26.38, 17.72, 28.73, 1.15, 1.94
	2nd vh	24.13, 31.08, 21.59, 33.62, 1.12, 1.08	26.62, 32.97, 23.79, 35.74, 1.57, 1.10	24.93, 32.33, 22.82, 34.52, 1.01, 1.34
	3rd vh	32.10, 39.48, 29.99, 42.12, 1.66, 1.70	29.57, 36.85, 27.41, 39.49, 1.36, 1.77	30.00, 37.04, 27.61, 39.98, 1.91, 1.21
2	1st vh	17.96, 25.35, 15.95, 27.50, 1.47, 1.62	18.45, 24.83, 15.80, 27.20, 1.89, 1.88	18.99, 25.78, 16.39, 28.29, 1.00, 1.87
	2nd vh	24.86, 31.19, 22.68, 33.38, 1.66, 1.96	23.86, 30.87, 21.02, 33.29, 1.88, 2.00	25.76, 32.85, 22.97, 35.10, 1.42, 1.00
	3rd vh	28.14, 35.84, 25.98, 38.35, 1.20, 1.63	28.96, 35.63, 26.26, 38.19, 1.48, 1.89	30.74, 37.21, 28.66, 39.64, 1.62, 1.76
3	1st vh	17.99, 24.79, 15.08, 27.75, 1.39, 1.20	17.28, 24.32, 14.51, 26.99, 1.19, 1.99	16.78, 23.72, 14.19, 26.44, 1.03, 1.49
	2nd vh	25.02, 31.91, 22.71, 34.74, 1.33, 1.87	22.93, 29.62, 20.30, 32.42, 1.10, 1.53	23.35, 29.85, 21.28, 32.75, 1.19, 1.36
	3rd vh	29.70, 36.69, 27.63, 39.18, 1.30, 1.52	30.58, 38.05, 28.56, 41.02, 1.06, 1.18	29.09, 36.33, 26.92, 38.57, 1.83, 1.71
4	1st vh	16.76, 23.43, 14.70, 25.87, 1.30, 1.76	17.03, 24.57, 14.67, 26.64, 1.35, 1.82	17.02, 24.43, 14.44, 26.47, 1.42, 1.13
	2nd vh	24.10, 30.90, 22.00, 33.02, 1.89, 1.80	24.38, 31.16, 21.76, 33.51, 1.53, 1.36	23.35, 30.48, 20.42, 32.77, 1.86, 1.05
	3rd vh	31.79, 38.68, 28.80, 41.06, 1.40, 1.54	29.41, 36.55, 27.05, 39.45, 1.02, 1.62	32.23, 39.54, 29.57, 42.52, 1.46, 1.37
5	1st vh	32.23, 39.54, 29.57, 42.52, 1.46, 1.37	20.27, 27.47, 17.44, 30.07, 1.39, 1.76	20.23, 26.60, 18.05, 29.09, 1.24, 1.16
	2nd vh	26.44, 33.54, 23.88, 36.17, 1.47, 1.54	25.25, 31.71, 23.17, 33.78, 1.79, 1.50	24.55, 32.08, 21.96, 34.78, 1.82, 1.23
	3rd vh	28.41, 35.46, 26.23, 38.06, 1.08, 1.22	32.56, 39.70, 29.96, 42.20, 1.22, 1.66	31.55, 38.72, 29.01, 41.03, 1.42, 1.10
6	1st vh	16.97, 23.32, 14.42, 25.82, 1.58, 1.51	19.39, 26.58, 17.30, 29.14, 1.04, 1.74	16.62, 23.44, 13.93, 26.09, 1.66, 1.63
	2nd vh	26.51, 32.93, 23.60, 35.07, 1.20, 1.95	22.75, 30.56, 20.49, 33.21, 1.94, 1.84	26.22, 32.99, 23.77, 35.00, 1.26, 1.23
	3rd vh	27.93, 34.97, 25.76, 37.40, 1.40, 1.71	30.06, 36.42, 27.92, 38.65, 1.84, 1.63	29.97, 36.90, 27.66, 39.16, 1.15, 1.12
7	1st vh	21.76, 28.85, 19.07, 30.96, 1.01, 1.94	18.45, 25.44, 15.77, 28.07, 1.65, 1.60	18.95, 26.34, 16.86, 29.02, 1.96, 1.34
	2nd vh	25.44, 32.52, 23.41, 34.71, 1.78, 1.20	23.54, 30.92, 21.46, 33.04, 1.25, 1.67	25.72, 32.41, 23.00, 34.47, 1.31, 1.41
	3rd vh	27.47, 35.33, 25.40, 37.66, 1.87, 1.15	29.96, 37.50, 27.54, 39.91, 1.83, 1.82	31.64, 38.63, 28.70, 41.38, 1.76, 1.42

References

Applegate D, Cook W, Dash S (2002) Solution of a min–max vehicle routing problem. Inf J Comput 14(2):132–143

Bektas T (2006) The multiple traveling salesman problem: an overview of formulations and solution procedures. Omega 34:209–219

Bertazzi L, Maggioni F (2014) Solution approaches for the stochastic capacitated traveling salesmen location problem with recourse. J Optim Theory Appl 166(1):321–342

Bigras LP, Gamache M, Savard G (2008) The time-dependent traveling salesman problem and single machine scheduling problems with sequence dependent setup times. Discrete Optim 5(4):685–699

Borne S, Mahjoub AR, Taktak R (2013) A branch-and-cut algorithm for the multiple steiner TSP with order constraints. Electron Notes Discrete Math 41(5):487–494

Carter AE, Ragsdale CT (2006) A new approach to solving the multiple traveling salesperson problem using genetic algorithms. Eur J Oper Res 175:246–257

Changdar C, Maiti MK, Maiti M (2013) A constrained solid tsp in fuzzy environment: two heuristic approaches. Iran J Fuzzy Syst 10(1):1–28

Colorni A, Dorigo M, Maniezzo V (1991) Distributed optimization by ant colonies. In: Proceedings of the first European conference on artificial life, Paris, pp 134–142

Colorni A, Dorigo M, Maniezzo V, Trubian M (1994) Ant system for job-shop scheduling. Belgian J Oper 34:39–53

Deer L, Verbiest N, Cornelis C, Godo L (2015) A comprehensive study of implicator–conjunctor-based and noise-tolerant fuzzy rough sets: definitions, properties and robustness analysis. Fuzzy Sets Syst 275(15):1–38

Dorigo M, Colorni A (1996) The ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern Part B 26(1):1–13

Dorigo M, Gambardella LM (1997) Ant colonies for the travelling salesman problem. Bio Syst 43(2):73–81

Dorigo M, Gambardella LM (1997) Ant colony system: a cooperative learning approach to the travelling salesman problem. IEEE Trans Evolut Comput 1(1):53–66

Dorigo M, Maniezzo V, Colorni A (1996) Ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern Part-B Cybern 26(1):29–41

Dubois D, Prade H (1987) Twofold fuzzy sets and rough sets—some issues in knowledge representation. Fuzzy Sets Syst 23(1):3–18

Dubois D, Prade H (1990) Rough fuzzy sets and fuzzy rough sets. Int J Gen Syst 17:191–208

Engelbrech AP (2005) Fundamentals of computational swarm intelligence. Wiley, New York

Gen M, Cheng R (1997) Genetic Algorithms and Engineering Design. Wiley, USA

Goldbarg MC, Bagi LB, Goldbarg EFG (2008) Transgenetic algorithm for the traveling purchaser problem. Eur J Oper Res 199:36–45

Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Boston

Gwiazda TD (2006) Genetic algorithms reference: crossover for single-objective numerical optimization problems, vol I. Springer, Berlin

Holland JH (1975) Adaptation in natural and artificial systems. MIT Press, Cambridge

Hsu C, Tsai M, Chen W (1991) A study of feature-mapped approach to the multiple travelling salesmen problem. IEEE Int Sympos Circuits Syst 3:1589–1592

Kiraly A, Christidou M, Chován T, Karlopoulos E, Abonyi J (2015) Minimization of off-grade production in multi-site multi-product plants by solving multiple traveling salesman problem. J Clean Prod (accepted manuscript, available online 19 May 2015)

Kota L, Jarmai K (2015) Mathematical modeling of multiple tour multiple traveling salesman problem using evolutionary programming. Appl Math Model 39(12):3410–3433

Liu B (2002) Theory and practice of uncertain programming. Physica, Heidelberg

Liu M, Chen D, Wu C, Li H (2006) Fuzzy reasoning based on a new fuzzy rough set and its application to scheduling problems. Comput Math Appl 51:1507–1518

Ma J, Li M, Zhang Y, Zhou H (2014) Firefly algorithm solving equal-task multiple traveling salesman problem. Commun Comput Inf Sci 472:304–307

Michalewicz Z (1992) Genetic algorithms + data structures = evolution programs. Springer, Berlin

Mladenović N, Urošević D, Hanafi S, Ilić A (2012) A general variable neighbourhood search for the one-commodity pickup-and-delivery travelling salesman problem. Eur J Oper Res 220(1):270–285

Ouaarab A, Ahiod B, Yang XS (2013) Discrete cuckoo search algorithm for the travelling salesman problem. Neural Comput Appl 24(7–8):1659–1669

Orloff DS (1974) Routing a fleet of M vehicles to/from a central facility. Networks 4:147–162

Pawlak Z (1982) Rough sets. Int J Comput Inf Sci 11(5):341–356

Pratihar DK (2008) Soft computing. Narosa Publishing House, New Delhi

Radzikowska AM, Kerre EE (2002) A comparative study of fuzzy rough sets. Fuzzy Sets Syst 126:137–156

Ryan JL, Bailey TG, Moore JT, Carlton WB (1998) Reactive tabu search in unmanned aerial reconnaissance simulations. The 30th conference on winter simulation. IEEE Computer Society Press, Los Alamitos, pp 873–879

Salazar-González J, Santos-Hernández B (2015) The split-demand one-commodity pickup-and-delivery travelling salesman problem. Transp Res Part B Methodol 75:58–73

Sarin SC, Sherali HD, Judd JD, Tsai P-F (2014) Multiple asymmetric traveling salesmen problem with and without precedence constraints: performance comparison of alternative formulations. Comput Oper Res 51:64–89

Shen Q, Richard J (2004) Selecting informative features with fuzzy-rough sets and its application for complex systems monitoring. Pattern Recognit 37:1351–1363

Tsai PF (2006) Tight flow-based formulations for the asymmetric traveling salesman problem and their applications to some scheduling problems. Ph.D. dissertation. Virginia Polytechnic Institute and State University, Blacksburg

Wang Y (2003) Mining stock price using fuzzy rough set system. Expert Syst Appl 24:3–23

Wang CY, Hu BQ (2015) Granular variable precision fuzzy rough sets with general fuzzy relations. Fuzzy Sets Syst 275(15):39–57

Xu J, Zhao L (2008) A class of fuzzy rough expected value multi-objective decision making model and its application to inventory problems. Comput Math Appl 56:2107–2119

Zadeh LA (1965) Fuzzy sets. Inf Control 8(3):338–353

Zhao J, Liu Q, Wang W, Wei Z, Shi P (2011) A parallel immune algorithm for traveling salesman problem and its application on cold rolling scheduling. Inf Sci 181 7(1):1212–1223