Transportation Planning and Technology, April 2008
Vol. 31, No. 2, pp. 135–152

Routledge
Taylor & Francis Group

ARTICLE

Mitigating Traffic Congestion: Solving the Ride-Matching Problem by Bee Colony Optimization

DUŠAN TEODOROVIĆ*,** & MAURO DELL' ORCO†

**University of Belgrade, Belgrade, Serbia; **Virginia Tech, Falls Church, VA, USA &*
†Technical University of Bari, Bari, Italy

(Received 11 May 2006; Revised 10 January 2008; In final form 18 January 2008)

ABSTRACT Urban road networks in many countries are severely congested. Expanding traffic network capacities by building more roads is very costly as well as environmentally damaging. Researchers, planners, and transportation professionals have developed various Travel Demand Management (TDM) techniques, i.e. strategies that increase travel choices to travelers. Ride sharing is one of the widely used TDM techniques that assumes the participation of two or more persons that together share a vehicle when traveling from few origins to few destinations. In ride-matching systems, commuters wishing to participate in ride sharing are matched by where they live and work, and by their work schedule. There is no standard method in the open literature to determine the best ride-matching method. In this paper, an attempt has been made to develop the methodology capable to solve the ride-matching problem. The proposed Bee Colony Optimization Metaheuristic is sufficiently general and could be applied to various combinatorial optimization problems.

KEY WORDS: Travel Demand Management; ride matching; Bee Colony Optimization

Correspondence Address: Dušan Teodorović, University of Belgrade, Vojvode Stepe 305, 11000 Belgrade, Serbia. Email: dusan@sf.bg.ac.yu

ISSN 0308-1060 print: ISSN 1029-0354 online
DOI: 10.1080/03081060801948027

Introduction

Urban road networks in many countries are severely congested, resulting in increased travel times, increased number of stops, unexpected delays, greater travel costs, inconvenience to drivers and passengers, increased air pollution and noise levels, and increased number of traffic accidents. Every day thousands of vehicles are delayed. Why do we wait? The answer is simple: demand that during certain time periods exceeds capacity is what results in queues. Service demand is characterized by the car arrival rate. Arrival rates significantly vary depending on the time of day, day in a week, or month in a year. Service rate indicates the number of cars that could be served in a given time unit. Obviously there are fluctuations in arrival rates and service times in many queueing systems. These fluctuations create queues and decrease the level of service offered to clients. The 'Hours of Delay per Traveler' measure is used to represent the congestion level in cities. It has been widely documented that traffic congestion has increased in cities of all sizes over the past two decades.

In order to achieve low increases in traffic congestion in cities, the rate of increasing supply should match the rate of increase in demand. Expanding traffic network capacities by building more roads is very costly as well as environmentally damaging. More efficient usage of the existing supply is essential in order to maintain the rising travel demand.

Researchers, planners, and transportation professionals have developed various Travel Demand Management (TDM) techniques (Vickrey, 1969; Yang & Huang, 1999; Phang & Toh, 2004; Sullivan & Harake, 1998; Teodorović & Edara, 2005), i.e. various strategies that increase travel choices to travelers. TDM strategies include alternative mode encouragement strategies such as 'Park-and-Ride facilities', 'High Occupancy Vehicle (HOV) facilities', 'Ride-sharing programs', 'Telecommuting', 'Alternative work hours', 'Congestion Pricing', 'Preferential parking to rideshare vehicles', among others.

In ride-matching systems, commuters wishing to participate in ride sharing are matched by where they live and work, and by their work schedule. There is no standard method in the open literature to determine the best ride-matching method. In this paper, an attempt has been made to develop the methodology capable to solve the ride-matching problem. The proposed methodology is based on the concepts of collective intelligence.

The paper is organized as follows: the ride-sharing concept is described in Section 'Ride Sharing', while Section 'The Ride-Matching Problem' contains the statement of the ride-matching problem. The proposed solution to the ride-matching problem is described in Section

'Proposed Solution to the Ride-Matching Problem'. The new computational paradigm – the Bee Colony Optimization (BCO) – is explained in Section 'The Bee Colony Optimization: The New Computational Paradigm'. Solving the Ride-Matching Problem by the Fuzzy Bee System (FBS) is described in Section 'Solving the Ride-Matching Problem by the Fuzzy Bee System'.

Ride Sharing

Ride sharing is one of the most widely used TDM techniques that assumes the participation of two or more persons that together share a vehicle when traveling from few origins to few destinations. The benefits of ride sharing are obvious: ride sharing significantly reduces the total number of trips. By sharing the ride with just one other commuter, one can decrease commuting everyday expenditure by 50%. At the same time, it is possible, while ride sharing, to use HOV lanes, to develop social life, and even create new friendships. Participants in ride sharing decide by themselves about various ride sharing operational issues (vehicle schedule, pick-up and drop-off points, maximum waiting time, music playing, smoking policy). Ride share programs use a wide range of traveler databases to match commuters who live and/or work in close proximity to each other for carpools and vanpools. Depending on the number of commuters in the group the *carpool* or the *vanpool* will be proposed and formed.

Carpooling is a widespread type of ride sharing. The participants in carpooling are neighbors who work at different companies located only a short distance away from each other, who also have similar work hours. The participants are frequently also staff of a single company who live next to each other. In some cases, the same traveler drives all the time, while the other commuters participate in sharing the cost. In some other cases, travelers alternate in driving.

Vanpooling is also a well-known type of ride sharing. A vanpool is usually composed of 5–15 commuters. Vans are leased or purchased by individuals that participate in vanpooling, by third party, or by employer, or a group of employers. The vanpool participants define the vanpool schedule and route. Most frequently, ride share programs put a new commuter into one of the vacant vanpools. The fares are based on the van type, and the mileage traveled.

The Ride-Matching Problem

There are various computerized ride-matching services that have been developed. In these systems, commuters wishing to participate in ride sharing are matched by where they live and work, and by their work schedule. Databases for ride matching are usually composed of hundreds or even thousands of names of commuters who want to share the ride. These bases contain information about names, telephones, home addresses, work addresses, work schedules, weekly frequencies of commuting (with specified commuting days), car availability, smoking preference, willingness to drive, etc.

On the other hand, there is no standard method in the open literature to determine the best ride matching. Obviously, there is a need to develop a more concrete methodology. The ride-matching problem considered in this paper could be defined in the following way: *Make routing and scheduling of the vehicles and passengers for the whole week in the* '*best possible way*'. All drivers that participate in ride sharing offer to the operator the following information regarding trips planned for the next week: (a) vehicle capacity (two, three, or four persons); (b) days in the week when person is ready to participate in ride sharing; (c) trip origin for every day in a week; (d) trip destination for every day in a week; and (e) desired departure and/or arrival time for every day in a week.

The following are potential objective functions: (a) minimize the total distance traveled by all participants; (b) minimize the total delay; and/or (c) make vehicle utilization relatively equal. We deal with the deterministic combinatorial optimization problem in the case when the desired departure and/or arrival times are fixed (for example, 'I want to be picked-up exactly at 8:00 a.m.'). On the other hand, in many real-life situations the desired departure and/or arrival times are *fuzzy* (I want to be picked-up about 8:00 a.m.). In this case, the ride-matching problem should be treated as a combinatorial optimization problem characterized by uncertainty.

Proposed Solution to the Ride-Matching Problem

The ride-matching problem could be treated as a deterministic combinatorial optimization problem, or as a combinatorial optimization problem characterized by uncertainty. In this paper, an attempt has been made to develop the methodology capable to solve both classes of the ride-matching problem. At the same time, the proposed methodology based on the collective intelligence concepts is sufficiently general and could be applied to various combinatorial optimization problems.

A great number of traditional engineering models and algorithms used to solve complex problems are based on control and centralization. Various natural systems (such as social insect colonies) lecture us that very simple individual organisms can create systems able to perform highly complex tasks by dynamically interacting with each other.

Bee swarm behavior in nature is, first and foremost, characterized by autonomy and distributed functioning and self-organizing. In the last couple of years, the researchers started studying the behavior of social insects in an attempt to use the Swarm Intelligence concept in order to develop various Artificial Systems.

The BCO Metaheuristic that represents the new direction in the field of Swarm Intelligence is introduced in this paper. One of the primary goals of this paper is to explore the possible applications of collective bee intelligence in solving combinatorial problems characterized by uncertainty. The development of the new heuristic algorithm for the ride-matching problem using the proposed approach shows the characteristics of the proposed concepts.

The Bee Colony Optimization (BCO): The New Computational Paradigm

Social insects (bees, wasps, ants, termites) have lived on Earth for millions of years, building nests and more complex dwellings, organizing production and procuring food. The colonies of social insects are very flexible and can adapt well to the changing environment. This flexibility allows the colony of social insects to be robust and maintain its life in spite of considerable disturbances.

The dynamics of the social insect population is a result of the different actions and interactions of individual insects with each other, as well as with their environment. The interactions are executed via a multitude of various chemical and/or physical signals. The final product of different actions and interactions represents social insect colony behavior. Interaction between individual insects in the colony of social insects has been well documented. The examples of such interactive behavior are bee dancing during the food procurement, ants' pheromone secretion, and performance of specific acts, which signal the other insects to start performing the same actions. These communication systems between individual insects contribute to the formation of the 'collective intelligence' of the social insect colonies. The term 'Swarm Intelligence' denoting this 'collective intelligence' has come into established use (Beni, 1988; Beni & Wang, 1989; Beni & Hackwood, 1992; Bonabeau *et al.*, 1999).

Bees in the Nature

Self-organization of bees is based on a few relatively simple rules of individual insect's behavior. In spite of the existence of a large number of different social insect species, and variation in their behavioral patterns, it is possible to describe individual insects' as capable of performing a variety of complex tasks (Camazine & Sneyd, 1991; Collevatti *et al.*, 1997; Dukas & Real, 1991; Dukas & Visscher, 1994; Gould, 1987; Kadmoon & Shmida, 1992; Seeley, 1992; Seeley & Visscher, 1988; Waddington *et al.*, 1998). The best example is the collection and processing of nectar, the practice of which is highly organized. Each bee decides to reach the nectar source by following a nestmate who has already discovered a patch of flowers. Each hive has a so-called dance floor area in which the bees that have discovered nectar sources dance, in that way trying to convince their nestmates to follow them. If a bee decides to leave the hive to get nectar, she follows one of the bee dancers to one of the nectar areas. Upon arrival, the foraging bee takes a load of nectar and returns to the hive relinquishing the nectar to a food storer bee. After she relinquishes the food, the bee can: (a) abandon the food source and become again an uncommitted follower; (b) continue to forage at the food source without recruiting the nestmates; or (c) dance and thus recruit the nestmates before the return to the food source. The bee opts for one of the above alternatives with a certain probability. Within the dance area, the bee dancers 'advertise' different food areas. The mechanisms by which the bee decides to follow a specific dancer are not well understood, but it is considered that 'the recruitment among bees is always a function of the quality of the food source' (Camazine & Sneyd, 1991). It is also noted that not all bees start foraging simultaneously. Experiments confirm that new bees begin foraging at a rate proportional to the difference between the eventual total and the number presently foraging.

Lučić and Teodorović (2001, 2003) were the first to use basic principles of collective bee intelligence in solving combinatorial optimization problems. They introduced the *Bee System* (*BS*) and tested it in the case of the Traveling Salesman Problem. The *BCO* Metaheuristic that has been proposed in this paper represents further improvement and generalization of the BS. The basic characteristics of the *BCO* Metaheuristic are described. Our artificial bee colony behaves partially alike, and partially differently from bee colonies in nature. The *FBS* (that represents special case of the *BCO*) capable to solve combinatorial optimization problems characterized by uncertainty is also introduced in the paper. Within *FBS*, the agents use approximate reasoning and rules of fuzzy logic (Zadeh, 1965, 1973; Teodorović & Vukadinović, 1998) in their communication and acting.

The Bee Colony Optimization (BCO) Metaheuristic

Within the *BCO* Metaheuristic, agents that we call *artificial bees* collaborate in order to solve difficult combinatorial optimization problems. All artificial bees are located in the hive at the beginning of the search process. During the search process, artificial bees communicate *directly*. Each artificial bee makes a series of local moves, and in this way incrementally constructs a solution to the problem. Bees are adding solution components to the current partial solution until they create one or more feasible solutions. The search process is composed of *iterations*. The first iteration is finished when bees create for the first time one or more feasible solutions. The best-discovered solution during the first iteration is saved, and then the second iteration begins. Within the second iteration, bees again incrementally construct solutions to the problem, *et seq*. There are one or more partial solutions at the end of each iteration. The analyst-decision maker prescribes the total number of iterations.

When flying through space our artificial bees perform either a *forward pass* or a *backward pass*. During a forward pass, bees create various partial solutions. They do this via a combination of individual exploration and collective experience from the past.

After that, they perform a *backward pass*, i.e. they return to the hive. In the hive, all bees participate in a *decision-making* process. We assume that every bee can obtain the information about solutions' quality generated by all other bees. In this way, bees *exchange* information about the quality of the partial solutions created. Bees compare all generated partial solutions. Based on the quality of the partial solutions generated, every bee decides whether to abandon the created partial solution and become again an uncommitted follower, continue to expand the same partial solution without recruiting nestmates, or dance and thus recruit the nestmates before returning to the created partial solution. Depending on the quality of the partial solutions generated, every bee possesses a certain level of *loyalty* to the path leading to the previously discovered partial solution. During the second forward pass, bees expand previously created partial solutions, and after that perform again the backward pass and return to the hive. In the hive bees again participate in a decision-making process, perform a third forward pass, etc. The iteration ends when one or more feasible solutions are created.

Like Dynamic Programming, the *BCO* also solves combinatorial optimization problems in stages. Each of the defined stages involves one optimizing variable. Let us denote by $ST = \{st_1, st_2, \ldots, st_m\}$ a finite set of pre-selected stages, where m is the number of stages. By B we denote the number of bees to participate in the search process, and by

I the total number of iterations. The set of partial solutions at stage st_j is denoted by S_j ($j = 1, 2, \ldots, m$).

The following is the pseudo-code of the BCO:

Bee Colony Optimization

---

(1) *Initialization*. Determine the number of bees B, and the number of iterations I. Select the set of stages $ST = \{st_1, st_2, \ldots, st_m\}$. Find any feasible solution x to the problem. This solution is the *initial best solution*.

(2) Set $i := 1$. Until $i = I$, repeat the following steps:

(3) Set $j = 1$. Until $j = m$, repeat the following steps:

Forward pass: Allow bees to fly from the hive and to choose B partial solutions from the set of partial solutions S_j at stage st_j.

Backward pass: Send all bees back to the hive. Allow bees to exchange information about quality of the partial solutions created and to decide whether to abandon the created partial solution and become again uncommitted followers, continue to expand the same partial solution without recruiting nestmates, or dance and thus recruit nestmates before returning to the created partial solution. Set, $j := j + 1$.

(4) If the best solution x_i obtained during the ith iteration is better than the best-known solution, update the best-known solution ($x := x_i$).

(5) Set, $i := i + 1$.

Alternatively, forward and backward passes could be performed until some other stopping condition is satisfied. The possible stopping conditions could be, for example, the maximum total number of forward/backward passes, or the maximum total number of forward/backward passes between two objective function value improvements.

Within the proposed *BCO* Metaheuristic, various sub-models describing bees' behavior and/or 'reasoning' could be developed and tested. In other words, various *BCO* algorithms could be developed. These models should describe the ways in which bees decide to abandon the created partial solution, to continue to expand the same partial solution without recruiting nestmates, or to dance and thus recruit nestmates before returning to the created partial solution.

In addition to proposing the *BCO* as a new metaheuristic, we also propose in this paper the *BCO* algorithm that we call the *FBS*. In the case of *FBS*, the agents (artificial bees) use approximate reasoning and rules of fuzzy logic in their communication and acting. In this way, the *FBS* is able to solve deterministic combinatorial problems, as well as combinatorial problems characterized by uncertainty.

The Fuzzy Bee System

Bees face many decision-making problems while searching for the best solution. The following are bees' choice dilemmas: (a) What is the next solution component to be added to the partial solution? (b) Should the partial solution be abandoned or not? (c) Should the same partial solution be expanded without recruiting nestmates?

The majority of the choice models are based on random utility modeling concepts. These approaches are highly rational. They are based on assumptions that decision makers possess perfect information processing capabilities and always behave in a rational way (trying to maximize utility). In order to offer an alternative modeling approach, researchers started to use less normative theories. The basic concepts of Fuzzy Set Theory, linguistic variables, approximate reasoning, and computing with words introduced by Zadeh (Beni & Hackwood, 1992; Bonabeau *et al.*, 1999) have more understanding for uncertainty, imprecision, and linguistically expressed observations. Following these ideas, we start in our choice model from the assumption that the quantities perceived by bees are 'fuzzy'. Artificial bees use approximate reasoning and rules of fuzzy logic (Zadeh, 1965, 1973; Teodorović & Vukadinović, 1998) in their communication and acting. During the jth stage bees fly from the hive and choose B partial solutions from the set of partial solutions S_i at stage st_j (forward pass). When adding the solution component to the current partial solution during the forward pass, a specific bee perceives a specific solution component as 'less attractive', 'attractive', or 'very attractive'. We also assume that an artificial bee can perceive a specific attribute as '*short*', '*medium*' or '*long*' (Figure 1), '*cheap*', '*medium*', or '*expensive*', *etc.*

Calculating the solution component attractiveness and choice of the next solution component to be added to the partial solution. The approximate reasoning algorithm for calculating the solution component attractiveness consists of the rules of the following type:

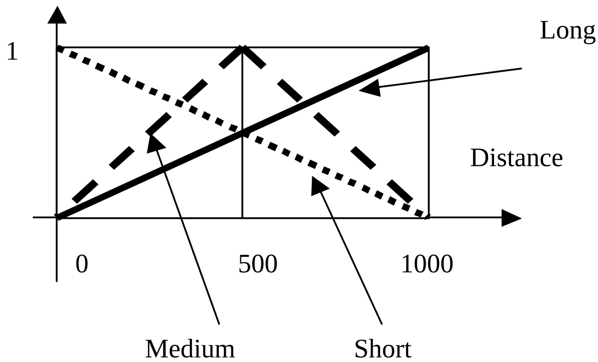

Figure 1. Fuzzy sets describing distance

If the attributes of the solution component are VERY GOOD
Then the considered solution component is VERY ATTRACTIVE

The main advantage of using the approximate reasoning algorithm for calculating the solution component attractiveness is that it is possible to calculate solution component attractiveness even if some of the input data were only *approximately* known. Let us denote by f_i the attractiveness value of solution component i. The probability p_i for solution component i to be added to the partial solution is equal to the ratio of f_i to the sum of all considered solution component attractiveness values:

$$p_i = \frac{f_i}{\sum_j f_i} \tag{1}$$

In order to choose the next solution component to be added to the partial solution, artificial bees use a proportional selection known as 'roulette wheel selection.' (The sections of roulette are in proportion to probabilities p_i). In addition to the 'roulette wheel selection,' several other ways of selection could be used.

Bee's partial solutions comparison mechanism. In order to describe bee's partial solutions comparison mechanism, we introduce the concept of *partial solution badness*. We define partial solution badness in the following way:

$$L_k = \frac{L^{(k)} - L_{\min}}{L_{\max} - L_{\min}} \tag{2}$$

where

L_k badness of the partial solution discovered by the kth bee

$L^{(k)}$ the objective function values of the partial solution discovered by the kth bee

$L_{\min}$ the objective function value of the best-discovered partial solution from the beginning of the search process

$L_{\max}$ the objective function value of the worst discovered partial solution from the beginning of the search process

The approximate reasoning algorithm to determine the partial solution badness consists of the rules of the following type:

If the discovered partial solution is BAD
Then loyalty is LOW

Bees use *approximate reasoning*, and compare their discovered partial solutions with the best, and the worst discovered partial solution from the *beginning* of the search process. In this way, 'historical facts' discovered by the *all members* of the bee colony have significant influence on the future search directions.

Bee's decision about recruiting nestmates. Since bees are, above all, social insects, it is assumed in this paper that the probability p^* of an event that a bee will continue to fly along the same path without recruiting nestmates is very low ($p^* \ll 1$). The bee flies to the dance floor, and starts dancing with the probability equal to $(1–p^*)$. This kind of communication between individual bees contributes to the formation of the 'collective intelligence' of the bee colony. In the case when a bee decides not to fly along the same path, the bee will go to the dancing area and will follow another bee(s).

Calculating the number of bees changing the path. Every partial solution (partial path) that is being advertised in the dance area has two main attributes: (a) the objective function value; and (b) the number of bees that are advertising the partial solution (partial path). The number of bees that are advertising the partial solution is a good indicator of a bees' collective knowledge. It shows how a bee colony perceives specific partial solutions.

The approximate reasoning algorithm to determine the advertised partial solution attractiveness consists of the rules of the following type:

If the length of the advertised path is SHORT and the number of bees advertising the path is SMALL
Then the advertised partial solution attractiveness is MEDIUM

Path attractiveness calculated in this way can take values from the interval [0,1]. The higher the calculated value, the more attractive is the advertised path. Bees are less or more loyal to 'old' paths. At the same time, advertised paths are less, or more, attractive to bees. Let us note paths p_i and p_j. We denote by n_{ij} the number of bees that will abandon path p_i, and join nestmates who will fly along path p_j.

The approximate reasoning algorithm to calculate the number of shifting bees consists of rules of the following type:

If bees' loyalty to path p_i is LOW and path p_j 's attractiveness is HIGH
Then the number of shifting bees from path p_i to path p_j is HIGH

In this way, the number of bees flying along a specific path is changed before beginning of the new forward pass. Using collective knowledge and sharing information among themselves, bees concentrate on more promising search paths, and slowly abandon less promising paths.

Solving the Ride-Matching Problem by the Fuzzy Bee System

Let us represent every passenger that participates in ride sharing by a node (Figure 2). We decompose our problem in stages. The first passenger in the car (driver) represents the first stage, the second passenger to join the ride sharing represents the second stage, the third passenger represents the third stage, et seq.

During a forward pass the bee will visit certain number of nodes, create a partial solution, and after that return to the hive (node O). In the hive the bee will participate in a decision-making process. Bees compare all generated partial solutions. Based on the quality of the partial solutions generated, every bee will decide whether to abandon the generated path and become again an uncommitted follower, continue to fly along a discovered path without recruiting nestmates, or dance and thus recruit nestmates before returning to the discovered path. Depending on the quality of the partial solutions generated, every bee possesses a certain level of loyalty to the path previously

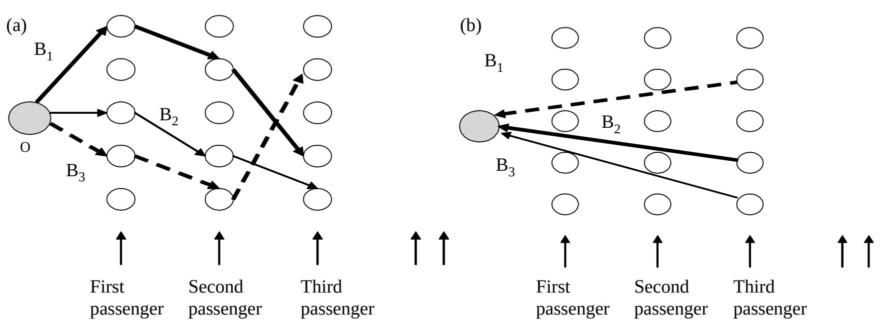

Figure 2. (a) First forward pass and (b) first backward pass

discovered. For example, bees B_1, B_2, and B_3 participated in the decision-making process. After comparing all generated partial solutions, bee B_1 decided to abandon an already generated path and join bee B_2.

The bees B_1 and B_2 fly together along the path generated by the bee B_2. When they reach the end of the path, they are free to make individual decisions about the next node to be visited. The bee B_3 will continue to fly along the discovered path without recruiting nestmates (Figure 3). In this way, bees are again performing a forward pass.

During the second forward pass, bees will visit a few more nodes, expand previously created partial solutions, and after that perform again the backward pass and return to the hive (node O). In the hive, bees will again participate in a decision-making process, make a decision, perform third forward pass, etc. The iteration ends when the bees have visited all nodes. When choosing the next node to be visited during the forward pass, the bee perceives a specific node as 'less attractive', 'attractive', or 'very attractive', depending on the proximity in space and proximity in time between two passenger requests. We call these proximities 'distance in space at origin', 'distance in space at destination', and 'distance of arrival times'.

We assume that an artificial bee can perceive a particular distance between nodes as '*short*', '*medium*' or '*long*'.

The approximate reasoning algorithm to determine the node attractiveness consists of the rules of the following type:

If the distance in space at origin is SHORT, and the distance in space at destination is SHORT, and the distance of arrival times is SHORT

Then the node attractiveness is HIGH

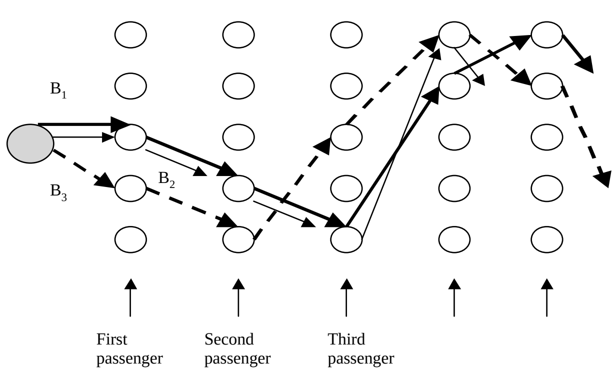

Figure 3. Second forward pass

The path badness (defined by eq. (2)) is used in the corresponding approximate reasoning algorithm to determine a bee's loyalty to the discovered path. The approximate reasoning algorithm to determine the advertised path attractiveness consists of rules of the following type:

If the length of the advertised path is SHORT, and the number of bees advertising the path is SMALL
Then the advertised path attractiveness is MEDIUM

Numerical Experiment

We tested the proposed model in the case of ride-sharing demand from Trani, a small attractive city in the south-east of Italy, to Bari, the regional capital of Puglia. We collected data regarding 97 travelers demanding ride sharing, and assumed, for sake of simplicity, that capacity is four passengers for all their cars. In our case, the algorithm chooses $24 \cdot 4 = 96$ out of 97 travelers to build up the 'best' path. In Figures 4 and 5 the travelers' origins and destinations are represented by diamonds. We used a hive of 15 bees, leaving at once. Only six 'foraging paths' have been completed by the bees; the other paths have been sooner or later abandoned.

The following are the best-discovered paths, and the corresponding clusters:

Optimal path = {21,63,17,66, 74,69,88,52, 36,77,92,61, 96,70,87,20, 53,34,93,29, 16,33,45,97, 27,76,2,11, 30,43,58,65, 19,26,39,60, 75,72,15,5, 86,81,13,6, 8,42,46,40, 56,32,24,83, 10,3,47,94, 25,89,91,49, 9,48,95,54, 68,31,71,50, 80,51,28,82, 44,64,57,59, 14,23,1,78, 22,67,79,18, 37,7,55,62, 90,84,41,85, 12,73,4,38}

Cluster #1 = {21,63,17,66}, Cluster #2 = {74,69,88,52}, Cluster #3 = {36,77,92,61}, Cluster #4 = {96,70,87,20}, Cluster #5 = {53,34,93,29}, Cluster #6 = {16,33,45,97}, Cluster #7 = {27,76,2,11}, Cluster #8 = {30,43,58,65}, Cluster #9 = {19,26,39,60}, Cluster #10 = {75,72,15,5}, Cluster #11 = {86,81,13,6}, Cluster #12 = {8,42,46,40}, Cluster #13 = {56,32,24,83}, Cluster #14 = {10,3,47,94}, Cluster #15 = {25,89,91,49}, Cluster #16 = {9,48,95,54}, Cluster #17 = {68,31,71,50}, Cluster #18 = {80,51,28, 82} Cluster #19 = {44,64, 57,59}, Cluster #20 = {14,23,1,78}, Cluster #21 = {22,67,79,18} Cluster #22 = {37,7,55,62}, Cluster #23 = {90,84,41,85}, Cluster #24 = {12,73,4,38}

Changes in the best-discovered objective function values are shown in Figure 6.

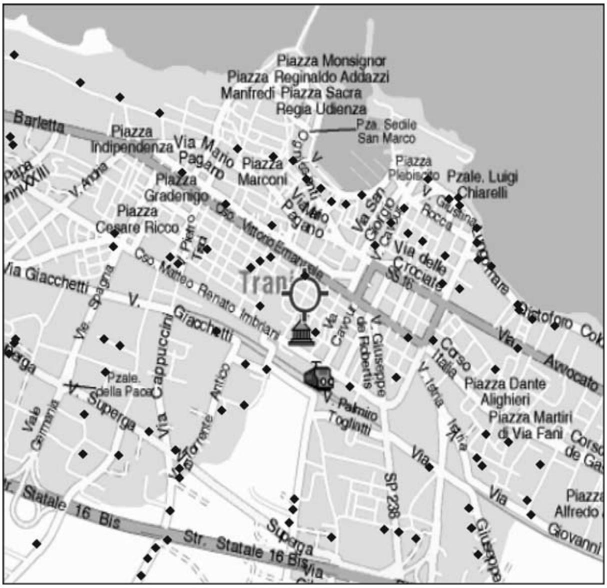

Figure 4. Location of origins

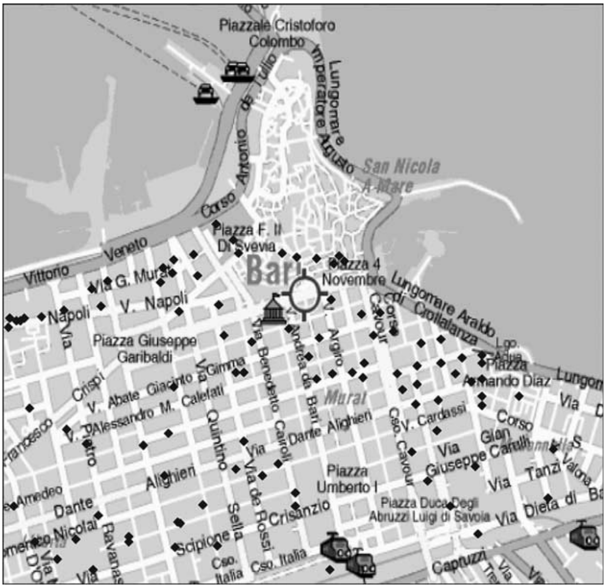

Figure 5. Location of destinations

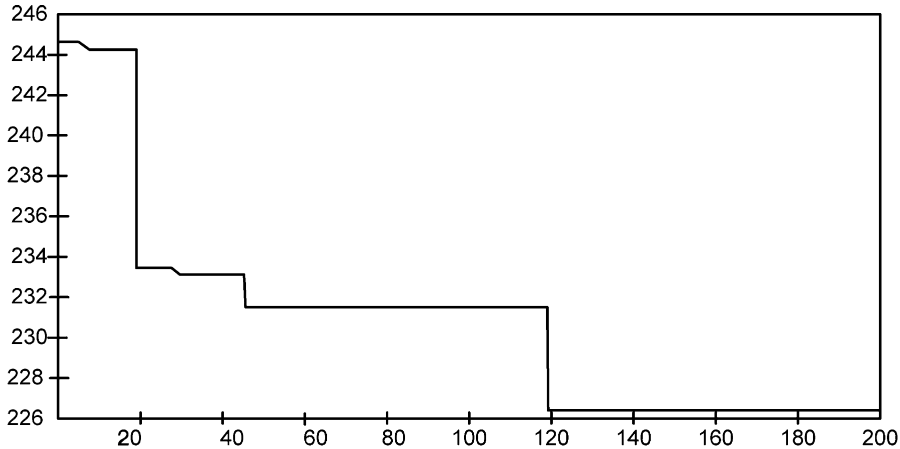

Figure 6. Changes in the best-discovered objective function values

Conclusions

Carpooling and vanpooling are widely used TDM strategies. They can significantly reduce the total number of trips in a network. Participants in ride sharing also save money, reduce stress, and reduce travel time since they can use HOV lanes. In ride-matching systems, commuters wishing to participate in ride sharing are matched by where they live and work, and by their work schedule. There is no standard method in the open literature to determine the best ride-matching method.

In this paper, an attempt has been made to develop the methodology to be able to solve the ride-matching problem. The proposed methodology was based on the concepts of collective intelligence. The proposed BCO Metaheuristic was sufficiently general and could be applied to various combinatorial optimization problems. There are, however, no theoretical results at this time that could support such a proposed approach. The development of the fuzzy rule basis and the choice of membership functions assume a trial-and-error procedure. Usually, the development of various metaheuristics was based on experimental work in the initial stages. Good experimental results usually motivate researchers to try to produce some theoretical results. The concepts proposed in this paper are not an exception in this sense.

Preliminary results of the BCO appear very promising. These results indicate that the development of new models based on swarm intelligence principles could significantly contribute to the solution of a wide range of complex engineering and management problems.

References

Beni, G. (1988) The concept of Cellular Robotic System, in: *Proceedings 1988 IEEE International Symposium on Intelligent Control*, Los Alamitos, CA: IEEE Computer Society Press, pp. 57–62.

Beni, G. & Hackwood, S. (1992) Stationary waves in cyclic swarms, in: *Proceedings 1992 International Symposium on Intelligent Control*, Los Alamitos, CA: IEEE Computer Society Press, pp. 234–242.

Beni, G. & Wang, J. (1989) Swarm Intelligence, in: *Proceedings Seventh Annual Meeting of the Robotics Society of Japan*, Tokyo: RSJ Press, pp. 425–428.

Bonabeau, E., Dorigo, M. & Theraulaz, G. (1999) *Swarm Intelligence* (Oxford, UK: Oxford University Press).

Camazine, S. & Sneyd, J. (1991) A model of collective nectar source by honey bees: Self-organization through simple rules, *Journal of Theoretical Biology*, 149, pp. 547–571.

Collevatti, R. G., Campos, L. A. O. & Schoereder, J. H. (1997) Foraging behaviour of bee pollinators on the tropical weed *Triumfetta semitriloba*: Departure rules from flower patches, *Insectes Sociaux*, 44, pp. 345–352.

Dukas, R. & Real, L. A. (1991) Learning foraging by bees: A comparison between social and solitary species, *Animal Behaviour*, 42, pp. 269–276.

Dukas, R. & Visscher, P. K. (1994) Lifetime learning by foraging honey bees, *Animal Behavior*, 48, pp. 1007–1012.

Gould, J. L. (1987) Landmark learning by honey bees, *Animal Behaviour*, 35, pp. 26–34.

Kadmoon, R. & Shmida, A. (1992) Departure rules used by bees foraging for nectar: A field test, *Evolutionary Ecology*, 6, pp. 142–151.

Lučić, P. & Teodorović, D. (2001) Bee system: Modeling combinatorial optimization transportation engineering problems by Swarm Intelligence, in: *The Preprints of the TRISTAN IV Triennial Symposium on Transportation Analysis*, Sao Miguel, Azores Islands, Portugal: University of Azores, pp. 441–445.

Lučić, P. & Teodorović, D. (2003) Computing with bees: Attacking complex transportation engineering problems, *International Journal on Artificial Intelligence Tools*, 12, pp. 375–394.

Phang, S. Y. & Toh, R. S. (2004) Road congestion pricing in Singapore: 1975 to 2003, *Transportation*, 43, pp. 16–25.

Seeley, T. D. (1992) The tremble dance of the honey bee: Message and meanings, *Behavior Ecology Sociobiology*, 31, pp. 375–383.

Seeley, T. D. & Visscher, P. K. (1988) Assesing the benefits of cooperation in honeybee foraging: Search costs, forage quality, and competitive ability, *Behavior Ecology Sociobiology*, 22, pp. 229–237.

Sullivan, E. C. & El Harake, J. (1998) California route 91 toll lanes – Impacts and other observations, *Transportation Research Record*, 1649, pp. 55–62.

Teodorović, D. & Edara, P. (2005) Highway Space Inventory Control System, in: H. Mahmassani (Ed.) *Transportation and Traffic Theory: Flow, Dynamics and Human Interaction* (Amsterdam, The Netherlands: Elsevier) pp. 43–62.

Teodorović, D. & Vukadinovi(, K. (1998) *Traffic Control and Transport Planning: A Fuzzy Sets and Neural Networks Approach* (Boston, MA: Kluwer Academic Publishers).

Vickrey, W. (1969) Congestion theory and transport investment, *American Economic Review (Papers and Proceedings)*, 59, pp. 251–261.

Waddington, K. D., Nelson, C. M. & Page, R. E. Jr. (1998) Effects of pollen quality and genotype on the dance of foraging honey bees, *Animal Behaviour*, 56, pp. 35–39.

Yang, H. & Huang, H. J. (1999) Carpooling and congestion pricing in a multilane highway with high-occupancy-vehicle lanes, *Transportation Research*, 33A, pp. 139–155.

Zadeh, L. (1965) Fuzzy sets, *Information and Control*, 8, pp. 338–353.

Zadeh, L. (1973) Outline of a new approach to the analysis of complex system and decision processes, *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3, pp. 28–44.