Expert Systems With Applications 66 (2016) 163–175

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

An ant colony system empowered variable neighborhood search algorithm for the vehicle routing problem with simultaneous pickup and delivery

Can B. Kalayci*, Can Kaya

Pamukkale University, Faculty of Engineering, Department of Industrial Engineering, Kinikli, 20070 Denizli, Turkey

ARTICLE INFO

Article history:
Received 8 January 2016
Revised 9 September 2016
Accepted 10 September 2016
Available online 10 September 2016

Keywords:
Vehicle routing problem
Simultaneous pickup and delivery
Time limit
Ant colony system
Variable neighborhood search
Metaheuristics

ABSTRACT

Along with the progress in computer hardware architecture and computational power, in order to overcome technological bottlenecks, software applications that make use of expert and intelligent systems must race against time where nanoseconds matter in the long-awaited future. This is possible with the integration of excellent solvers to software engineering methodologies that provide optimization-based decision support for planning. Since the logistics market is growing rapidly, the optimization of routing systems is of primary concern that motivates the use of vehicle routing problem (VRP) solvers as software components integrated as an optimization engine. A critical success factor of routing optimization is quality vs. response time performance. Less time-consuming and more efficient automated processes can be achieved by employing stronger solution algorithms. This study aims to solve the Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRPSPD) which is a popular extension of the basic Vehicle Routing Problem arising in real world applications where pickup and delivery operations are simultaneously taken into account to satisfy the vehicle capacity constraint with the objective of total travelled distance minimization. Since the problem is known to be NP-hard, a hybrid metaheuristic algorithm based on an ant colony system (ACS) and a variable neighborhood search (VNS) is developed for its solution. VNS is a powerful optimization algorithm that provides intensive local search. However, it lacks a memory structure. This weakness can be minimized by utilizing long term memory structure of ACS and hence the overall performance of the algorithm can be boosted. In the proposed algorithm, instead of ants, VNS releases pheromones on the edges while ants provide a perturbation mechanism for the integrated algorithm using the pheromone information in order to explore search space further and jump from local optima. The performance of the proposed ACS empowered VNS algorithm is studied on well-known benchmarks test problems taken from the open literature of VRPSPD for comparison purposes. Numerical results confirm that the developed approach is robust and very efficient in terms of both solution quality and CPU time since better results provided in a shorter time on benchmark data sets is a good performance indicator.

1. Introduction

Vehicle routing problem (VRP) variants arising in real world applications are considered to be highly complex combinatorial optimization problems that require improved automation and high-quality software tools for decision support since human planning is inadequate for most applications and occupation of valuable human resources is undesired. Computerized planning not only reduces risk of error, but also improves the utilization of transportation resources and provides more efficient planning processes with respect to manual planning. As efficiency and sustainability grows in importance, significant cost savings and a better utilization of resources serving to environmental benefits can be achieved via the combination of powerful computers and high performance solvers integrated to expert and intelligent software systems. A key strength of excellent solvers comes from the better optimization performance yielding to better solutions with substantial savings. Gap between the requirements and the performance of today's optimization-based decision support systems is quite large. The evolving optimization technology of tomorrow may be greatly enhanced by improved solution algorithms that play an essential role in finding high-quality feasible solutions even for more complex and larger sized real-life instances. Therefore, there is a huge potential of improvement and hence an exciting work ahead.

* Corresponding author: Fax: +90 258 296 3262.
E-mail address: cbkalayci@pau.edu.tr (C.B. Kalayci).

http://dx.doi.org/10.1016/j.eswa.2016.09.017
0957-4174/

Companies must wisely optimize their decisions in every aspect in order to survive in today's competitive environment. The optimization of logistic systems is of primary and shared concern for many companies since it serves not only to reduced transportation costs and improved service quality, but also to environmental protection. A critical step of this optimization process is to solve the vehicle routing problem (VRP) introduced by Dantzig and Ramser (1959). The VRP that is concerned with the optimal set of routes for a fleet of vehicles to traverse in order to deliver to a given set of clients. A widely studied generalization of VRP, the vehicle routing problem with simultaneous pickup and delivery (VRPSPD), proposed by Min (1989), requires simultaneous consideration of both pickup and delivery demands. This causes a fluctuation in the current load of the vehicle which results in increased difficulty in checking the feasibility of the solutions. Therefore, a key aspect is to check the current load of the vehicle(s) at each client since the vehicle capacity cannot not be exceeded. A well-known example occurs in the soft drink industry where the operations of delivering full bottles and picking empty ones up are performed by the same vehicle. A further extension of VRPSPD, vehicle routing problem with simultaneous pickup and delivery with time limit (VRPSPDTL), additionally requires the vehicles to return to the central depot before a time deadline is reached. A real life example occurs in milk transportation since there is a limited time to carry such sensitive goods on the route before turning bad. Mathematical formulation of VRPSPD and VRPSPDTL can be found in Alfredo Tang Montané and Galvão (2006) and Polat, Kalayci, Kulak, and Günther (2015), respectively.

Since VRPSPD works were started by Min (1989) with a case study of a small sized book distribution problem that requires book delivery and pick-up operations between a central library and twenty-two local libraries, several studies have been published on this problem. These studies in the literature can be categorized as exact, heuristic, single solution based and population based metaheuristic algorithms.

Since the core problem is known to be NP-hard, very few exact approaches have been developed for this problem. The first exact solution methodology based on branch-and-price approach with exact dynamic programming and state space relaxation procedures was developed by Dell'Amico, Righini, and Salani (2006) in which only instances up to 40 customers could be solved to optimality despite high computational time. Other exact solution approaches have been proposed by Subramanian, Uchoa, Pessoa, and Ochi (2011) based on branch-and-cut algorithm and Subramanian, Uchoa, Pessoa, and Ochi (2013) based on branch-cut-and-price algorithm which were able to solve instances up to 100 clients. Although exact methodologies are very useful to prove optimal solutions and to provide new lower bounds for well-known data instances, since the solution space grows exponentially when the problem size increases due to computational complexity (Tovey, 2002) which has a significant influence on the running time of algorithms, exact solution procedures may not be adequately efficient within reasonable time. Therefore, heuristic and metaheuristic approaches are commonly developed as solution methodologies.

Dethloff (2001) proposed a mathematical formulation for VRPSPD and developed an insertion-based heuristic by addressing the problem in reverse logistics operations. Salhi and Nagy (1999) studied a similar approach that combined Clarke and Wright saving heuristic (Clarke & Wright, 1964) with an insertion based heuristic that insert two customers at a time in the route. Nagy and Salhi (2005) developed a composite heuristic approach that temporarily allows a certain degree of infeasibility to occur at VRPSPD solutions and eliminates capacity infeasibilities using different sub-routines such as 2-opt, 3-opt, shift, exchange, reverse and perturb structures adapted for the problem. Heuristic approaches usually do not include an improvement routine and thus ends up with the same result for every run of the procedure unless a probabilistic mechanism is used. Therefore, heuristics are either used for initial solution construction in order to provide a better starting point compared to pure randomness or as an embedded constructive sub procedure of iterative search algorithms.

Single solution based metaheuristic algorithms have been commonly used for solving VRPSPD. Among the existing solution strategies, tabu search method is by far the most preferred approach applied for solving this problem. The first tabu search study is proposed by Crispim and Brandao (2005) which is a combination of tabu search and variable neighborhood descend approach with a sweep procedure that allows infeasibility to construct initial solution as well as insert and swap procedures to improve the incumbent solution until feasibility is established by penalizing according to the level of overloads. Chen and Wu (2005) proposed a tabu search algorithm that obtains initial solutions by an insertion-based procedure based on distance and load based criteria and then improves solutions with 2-exchange, swap, shift, 2-opt and Or-opt procedures and record-to-record travel strategies. Alfredo Tang Montané and Galvão (2006) proposed a tabu search algorithm that makes use of inter-route and intra-route neighborhood structures such as interchange, relocation, crossover and 2-opt procedures by controlling intensification and diversification scheme of the approach with a frequency penalization scheme. Bianchessi and Righini (2007) also proposed a tabu search algorithm with a variable neighborhood structure that makes use of node and arc exchange based local search heuristics. Wassan, Wassan, and Nagy (2008) proposed a tabu search algorithm with a mechanism that dynamically controls the tabu list size to achieve an effective balance between the intensification and diversification of the conducted search and makes use of neighborhood structures such as shift, swap, local shift and reverse procedures for improvement. Zachariadis, Tarantilis, and Kiranoudis (2009) presented a tabu search algorithm combined with guided local search strategies that iteratively improves the initial solution generated by a saving based constructive heuristic using neighborhood structures such as customer relocation, customer exchange, route interchange procedures. Zachariadis, Tarantilis, and Kiranoudis (2010) proposed a solution approach that stores the routes of high quality VRPSPD solutions in memory and makes use of an improvement procedure based on tabu search. Zachariadis and Kiranoudis (2011) utilize the weighted savings based heuristic approach to generate an initial solution and then a tabu search algorithm to improve the solutions. Ropke and Pisinger (2006) developed a large neighborhood search approach that selects a neighborhood method according to a probability depending on its success for solving several variants of vehicle routing problems with backhauls involving VRPSPD. A multi-start metaheuristic approach which consists of a variable neighborhood descent procedure, a random neighborhood ordering procedure and an iterated local search (ILS) framework is proposed by Subramanian, Drummond, Bentes, Ochi, and Farias (2010) who performed the experiments in a cluster with a multicore architecture using up to 256 cores utilizing the parallel structure of the algorithm. Jun and Kim (2012), Li, Pardalos, Sun, Pei, and Zhang (2015) and Avci and Topaloglu (2015) proposed iterated local search algorithms with inter-route and intra-route operators and perturbation mechanisms to solve VRPSPD. Polat, Kalayci, Kulak, et al. (2015) recently proposed a perturbation based variable neighborhood search algorithm for solving VRPSPD with and without time limit restrictions.

Population based metaheuristic algorithms have also been applied for solving VRPSPD. Ai and Kachitvichyanukul (2009) and Goksal, Karaoglan, and Altiparmak (2013) proposed particle swarm optimization algorithms for solving VRPSPD. While Ai and Kachitvichyanukul (2009) used a real value encoding mechanism for representation of the solution, cheapest insertion heuristic and 2-

Table 1
Literature summary for VRPSPD.

Author(s)	Category	Method
Min (1989)	H	Problem introduction with a case study
Salhi and Nagy (1999)	H	IBH: Insertion-based heuristic
Dethloff (2001)	H	IBH: Insertion-based heuristic
Nagy and Salhi (2005)	H	CH: Composite heuristic
Crispim and Brandao (2005)	SSBM	TSVND: Tabu search and variable neighborhood descent
Chen and Wu (2005)	SSBM	TS: Tabu search
Dell'Amico et al. (2006)	E	BAP: Branch-and-price algorithm
Alfredo Tang Montané and Galvão (2006)	SSBM	TS: Tabu search
Ropke and Pisinger (2006)	SSBM	LNS: Large Neighborhood Search
Bianchessi and Righini (2007)	SSBM	TS: Tabu search
Wassan et al. (2008)	SSBM	RTS: Reactive tabu search
Gajpal and Abad (2009)	PBM	ACS: Ant colony system
Ai and Kachitvichyanukul (2009)	PBM	PSO: Particle swarm optimization
Zachariadis et al. (2009)	SSBM	TSGLS: Tabu search and guided local search
Çatay (2010)	PBM	SBAA: Saving based ant algorithm
Zachariadis et al. (2010)	SSBM	VLBR: Variable Length Bone Route
Subramanian et al. (2010)	SSBM	PILS: Parallel iterated local search
Zachariadis and Kiranoudis (2011)	SSBM	TS: Tabu search
Subramanian et al. (2011)	E	BAC: Branch-and-cut algorithm
Jun and Kim (2012)	SSBM	ILS: Iterated Local Seach
Tasan and Gen (2012)	PBM	GA: Genetic Algorithm
Subramanian et al. (2013)	E	BCAP: Branch-cut-and-price algorithm
Goksal et al. (2013)	PBM	HPSO: Hybrid particle swarm optimization
Polat, Kalayci, Kulak, et al. (2015)	SSBM	PVNS: Perturbation based variable neighborhood search
Avci and Topaloglu (2015)	SSBM	ALS: Adaptive local search
Li et al. (2015)	SSBM	ILSANS: Iterated local search embedded adaptive neighborhood selection

E: exact; H: heuristic; SSBM: single solution based metaheuristic; PBM: population based metaheuristic.

opt procedures for improvement, Goksal et al. (2013) implemented permutation encoding to represent a solution of the problem and a variable neighborhood descent algorithm as a local search to improve the randomly selected particles in each iteration. Tasan and Gen (2012) applied genetic algorithm based metaheuristic approach to VRPSPD. Ant colony optimization algorithms for solving VRPSPD are proposed in the studies of Gajpal and Abad (2009) and Çatay (2010) that both make use of Clark and Wright's savings algorithm (Clarke & Wright, 1964) and local search strategies such as 2-opt, customer insertion, interchange and sub-path exchange procedures.

It is worth to mention that since Salhi and Nagy (1999) defined some benchmark problem instances for VRPSPDTL that impose time limit restrictions to VRPSPD, it has received less attention in the literature. Among the studies mentioned above, the ones which consider time limit restrictions for VRPSPD instances are as follows: Ai and Kachitvichyanukul (2009), Alfredo Tang Montané and Galvão (2006), Çatay (2010), Dethloff (2001), Gajpal and Abad (2009), Nagy and Salhi (2005), Ropke and Pisinger (2006), Wassan et al. (2008), Jun and Kim (2012), Li et al. (2015), Polat, Kulak, and Günther (2012) and Polat, Kalayci, Kulak, et al. (2015).

The interested reader is referred to the review paper of Berbeglia, Cordeau, Gribkovskaia, and Laporte (2007) for more information about the vehicle routing problem with pickup and delivery as well as appropriate methods used to tackle this problem and its variants.

A brief review of the research literature of the VRPSPD with and without time limit restrictions is presented in Table 1 while a quantitative look on the metaheuristic strategies developed for solving VRPSPD is presented in Table 2.

As summarized in Table 2, single solution based metaheuristics are preferred as solution approaches over population based metaheuristics. There was only one strategy that takes advantage of parallel configuration while remaining algorithms are executed in sequential. Almost in each study, except a few, intra-route and inter-route neighborhood structures are employed. Although tabu search algorithm is by far the most investigated solution approach among other metaheuristics, in recent years, it appears that algorithms that iteratively employ intensive local search strategies such as large neighborhood search, variable neighborhood search and iterative local search solution approaches have gained popularity seeking an improvement in the solution quality vs. time performance.

Table 3 presents the algorithmic features and key strategies of metaheuristic algorithms proposed for solving VRPSPD. Initial solution construction strategy may provide a better start point for a solution approach. Sweep based and savings based algorithms are widely preferred for initialization in metaheuristics strategies. As iterative solution approaches, metaheuristics seek a better solution over generations in the vast solution space. The power of a metaheuristic algorithm lies in the fact that how effectively intensification and diversification strategies are employed. The decision between global search reinforcement and convergence search in the promising region is a key aspect of an efficient algorithm since a proper balance between intensification and diversification is desired. Therefore, metaheuristic strategies for VRPSPD are analyzed in detail according to their similarities and differences with a focus on intensification and diversification. It appears that variable neighborhood descent is a very popular procedure employed in the intensification phase of many algorithms. Taking advantage of well-known intra-route and inter-route neighborhood structures, single solution based metaheuristics are generally successful in achieving a good intensification while population based metaheuristics are generally better in diversification since many solutions are simultaneously investigated. Among the recent studies, the attention is given on different diversification strategies while similar intensification strategies are observed. Therefore, especially for single solution based metaheuristics, there is a logical reason to think that seeking of an efficient diversification mechanism comes into prominence in recent years. This paper describes a hybrid algorithm that combines the global search capability of any colony system and the intensification strategies of variable neighborhood search algorithm.

Table 2
Quantitative look on the metaheuristic strategies developed for solving VRPSPD.

#	Author(s)	Metaheuristic Category		Process Execution		Employed Neighborhood Structures		Main solution approach								
		SSBM	PBM	Sequential	Parallel	Intra-route	Inter-route	TS	VND	LNS	ACS	PSO	ILS	GA	VNS	ALS
1	Crispim and Brandao (2005)	×		×		×	×	×	×							
2	Chen and Wu (2005)	×		×		×	×	×								
3	Alfredo Tang Montané and Galvão (2006)	×		×		×	×	×								
4	Ropke and Pisinger (2006)	×		×		×	×			×						
5	Bianchessi and Righini (2007)	×		×		×	×	×								
6	Wassan et al. (2008)	×		×		×	×	×								
7	Gajpal and Abad (2009)		×	×		×	×				×					
8	Ai and Kachitvichyanukul (2009)		×	×								×				
9	Zachariadis et al. (2009)	×		×		×	×	×								
10	Çatay (2010)		×	×		×	×				×					
11	Zachariadis et al. (2010)	×		×		×	×	×								
12	A. Subramanian et al. (2010)	×			×	×	×						×			
13	Zachariadis and Kiranoudis (2011)	×		×		×	×	×								
14	Jun and Kim (2012)	×		×		×	×						×			
15	Tasan and Gen (2012)		×	×										×		
16	Goksal et al. (2013)		×	×		×	×					×				
17	Polat, Kalayci, Kulak, et al. (2015)	×		×		×	×								×	
18	Avci and Topaloglu (2015)	×		×		×	×									×
19	Li et al. (2015)	×		×		×	×						×			
	∑	14	5	18	1	17	17	8	1	1	2	2	3	1	1	1

SSBM: single solution based metaheuristic; PBM: population based metaheuristic.
TS: Tabu Search; VND: Variable Neighborhood Descent; LNS: Large Neighborhood Search; ACS: Ant Colony System; PSO: Particle Swarm Optimization; ILS: Iterated Local Search.
GA: Genetic Algorithm; VNS: Variable Neighborhood Search; ALS: Adaptive Local Search.

Expert and intelligent systems can gain significant success by incorporating efficient optimization tools that can enhance the optimizating power of a rule-based expert knowledge in a decision sup-port system. In this paper, our purpose is to present a fast solution approach that effectively solves VRPSPD with and without time limit restrictions. The solution approach is a hybridization of the ant colony system (ACS) and variable neighborhood search (VNS) which are well known metaheuristics in the literature. In the literature, since its first introduction by Dorigo and Gambardella (1997), ACS is hybridized with local search methodologies where local search is embedded to work as a sub procedure of ACS algorithm. In this study, on the contrary, ACS works as a sub procedure of VNS algorithm to empower VNS which does not have a memory structure. In the proposed algorithm, instead of ants, VNS releases pheromones on the edges while ants provide a perturbation mechanism as a diversification strategy for the integrated algorithm using the pheromone information in order to explore search space further. To the best of our knowledge, this is the first implementation of an integrated approach that combines ACS and VNS for solving VRPSPD and VRPSPTL. The proposed ant colony system empowered variable neighborhood search (ACOEVNS) algorithm is compared with the existing solution approaches for the VRPSPD and its extension with the maximum time limit constraint case, i.e. VRPSPDTL, in the literature. The computational results reveal that the ACOEVNS competes with existing approaches such as VNS approach that utilize a simple perturbation mechanism and ACS that utilize local search strategies in the literature very well and also improves some best known solutions of the benchmark problems studied.

The remainder of the paper is organized as follows: Section 2 presents the notation used throughout the paper. Section 3 presents the insights of the idea for developing a hybrid approach by combining two successful metaheuristics. Details of the proposed solution approach based on ant colony system and variable neighborhood search is presented in Section 4. Section 5 reports and discusses computational results for testing the performance of the developed method over publicly available benchmarks test problems. Section 6 presents a discussion regarding advantages and limitations of the proposed algorithm as well as insightful implications based on experimental outcomes. Finally, Section 7 discusses the contribution of the paper and concludes with directions for future research.

2. Notation

The notation used in this paper is listed below:

a	Indices for neighborhood structures ($a = 1, \ldots, a_{max}$)
b	Indices for neighborhood structures ($b = 1, \ldots, b_{max}$)
k	Indices for ants
r	Indices for nodes (clients or cities)
s	Indices for nodes (clients or cities)
u	Indices for nodes (clients or cities)
q	A random number uniformly distributed in [0,1]
q_0	Ant colony rule application parameter selected in [0,1]
$\tau(r, s)$	Desirability measure, that is the amount of pheromone accumulated on the edge (r, s)
$J_k(r)$	the set of clients (cities) that remain to be visited by ant k positioned on client r, to make the solution feasible

(*continued on next page*)

Table 3
Algorithmic features and key strategies of metaheuristic algorithms proposed for solving VRPSPD.

Author(s)	Initial solution construction strategy	Diversification strategy (reinforcement of global search)	Intensification strategy (reinforcement of convergence search in promising region)
Crispim and Brandao (2005)	Sweep based algorithm	Global tabu list	Dynamic VND
Chen and Wu (2005)	Insertion based heuristic	Tabu list	Local search with inter and intra route structures
Alfredo Tang Montané and Galvão (2006)	Tour partitioning heuristic and sweep based algorithm	Tabu tenure values controlled by a frequency-based memory procedure for a proper balance between intensification and diversification	
Ropke and Pisinger (2006)	Insertion heuristics	Choice of neighborhoods via employed heuristics maintain a proper balance between intensification and diversification.	
Bianchessi and Righini (2007)	Random Hamiltonian tour	Adaptive tuning of the tabu lists	TS with inter and intra route structures
Wassan et al. (2008)	Sweep based algorithm	Dynamically controlling the tabu list size	TS with inter and intra route structures
Gajpal and Abad (2009)	Nearest neighborhood heuristic	Probabilistic selection strategy of ACS	Pheromone release and local search neighborhood structures
Ai and Kachitvichyanukul (2009)	Random generation	Social Experience of the swarm	Cognitive experience of the particle
Zachariadis et al. (2009)	Weighted savings based algorithm	GLS with penalty function	TS with inter and intra route structures
Çatay (2010)	Nearest neighborhood heuristic	Probabilistic selection strategy of ACS	Pheromone release and local search neighborhood structures
Zachariadis et al. (2010)	Weighted savings based algorithm	Adaptive memory methodology that probabilistically generates partial solutions	Tabu search
A. Subramanian et al. (2010)	Random generation ensuring feasibility using greedy insertion strategy	Perturbation mechanism with inter-route neighborhood structures	Randomized VND
Zachariadis and Kiranoudis (2011)	Savings based algorithm	Aspiration criteria of tabu search	Promises mechanism using inter and intra route structures
Jun and Kim (2012)	Sweep Nearest Algorithm	Perturbation with destroy and repair	Randomly apply inter-route structures
Tasan and Gen (2012)	Random generation	Mutation	Crossover
Goksal et al. (2013)	Nearest Neighborhood Heuristic and random generation	Annealing process using cooling temperature	Sequential VND
Polat, Kalayci, Kulak, et al. (2015)	Savings based algorithm	Perturbation mechanism with inter-route neighborhood structures	Sequential VND
Avci and Topaloglu (2015)	Random generation ensuring feasibility	Adaptive Perturbation mechanism with inter-route neighborhood structures	Sequential VND
Li et al. (2015)	Savings based algorithm	Perturbation mechanism with LNS	Adaptive neighborhood selection mechanism

VND: Variable Neighborhood Descent; LNS: Large Neighborhood Search; ACS: Ant Colony System; GLS: Guided Local Search.

β	a parameter which determines the relative importance of pheromone versus heuristic information ($\beta > 0$)
ρ	Pheromone decay parameter
f_{gb}	Objective value of the global best tour (network)
N_a	The set of neighborhoods used in the VNS algorithm ($a = 1, \ldots, a_{max}$)
x^0	Initial solution for VNS algorithm
x	Incumbent solution of VNS algorithm
ilimit	Iteration limit to call ACS algorithm for perturbation
n	Number of clients of data instance
$\Delta\tau(r,s)$	Total pheromone accumulated on the edges between nodes r and s

3. Ant colony optimization and variable neighborhood search

In this section, we briefly give background information about the ACS optimization and VNS metaheuristics which are the two main contributors of the proposed integrated algorithm to solve the vehicle routing problem with simultaneous pickup and delivery with and without time limit. Then, hybridization ideas of the two methods is presented.

3.1. Ant colony system (ACS)

Ant colony system (ACS) is a distributed metaheuristic algorithm that is originally developed by Dorigo and Gambardella (1997) to solve travelling salesman problem (TSP) and its variants. In ACS, ants deposit pheromone on their path while moving from one position to another, thus an indirect, yet useful form of communication is established among the ant colony. Pheromone placed by the ants while on the go plays the role of a distributed long term memory although the memory is not stored locally within the individual ants, but is distributed on the path. Ants construct a solution to the problem by probabilistically selecting to move from one client (city) to another using the formula given by Eq. (1) and taking advantage of accumulated pheromone information calculated by Eq. (2).

$$p_k(r,s) = \begin{cases} \arg max_{u \in J_k(r)} \left\{ [\tau(r,u)] \cdot [\eta(r,u)]^{\beta} \right\} & if\ q \le q_0 \quad \text{(exploitation)} \\ \frac{[\tau(r,s)] \cdot [\eta(r,s)]^{\beta}}{\sum_{u \in J_k(r)} [\tau(r,u)] \cdot [\eta(r,u)]^{\beta}} & else\ if\ s \in J_k(r) \quad \text{(biased exploration)} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$\tau(r,s) \leftarrow (1-\rho) \cdot \tau(r,s) + \rho \cdot \Delta\tau(r,s)$$
$$where\ \Delta\tau(r,s) = \begin{cases} (f_{gb})^{-1} & if\ (r,s) \in \text{global best tour} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Since its successful application on TSP, ant colony algorithms have been adapted to solve various combinatorial optimization problems that take roots in TSP. Among those problems, vehicle routing problem (VRP) is, perhaps, the one which has the most several characteristics in common as it generalizes TSP with add on constraints. In the VRP literature, several applications of ant

```
select N_a, a = 1, ..., a_max
find x^0
choose a termination condition
x ← x^0
While termination condition not met do
    Set a ← 1
    While a < a_max do
        x' ← shake(N_a(x))
        x'' ← local search(x')
        If move Or Not (x, x'') then
            x ← x''
            a ← 1
            If f(x) < f(x^0) then
                x^0 ← x
            End if
        Else
            a ← a + 1
        End if
    End while
End while
```

Fig. 1. Steps of the basic VNS algorithm based on Hansen and Mladenović (2001).

colony algorithms can be noticed. Ant algorithms are adapted to solve VRPSPD with promising results by Çatay (2010) and Gajpal and Abad (2009) using an embedded savings based heuristic information that is borrowed from Clarke and Wright (1964).

3.2. Variable neighborhood search (VNS)

Proposed by Mladenović and Hansen (1997), variable neighborhood search (VNS) approach, a local search algorithm, employs the idea of systematically changing of neighborhoods in order to explore increasingly distant neighborhoods and improve current incumbent solution by jumping from one solution to another. Thus, favorable characteristics of the incumbent solution is preserved. In recent years, VNS received more attention in the VRP literature. Successful applications of VNS for VRP variants are as follows: Vehicle routing problem with deliveries and selective pickups (Coelho et al., 2015), vehicle network design problem (Polat, Gunther, & Kulak, 2014), capacitated vehicle routing problem with two-dimensional loading constraints (Wei, Zhang, Zhang, & Lim, 2015) and VRPSPDTL (Polat, Kalayci, Kulak, et al., 2015). Polat, Kalayci, Kulak, et al. (2015) pointed out the necessity of a perturbation mechanism for VNS algorithms. Some other applications and extensions of VNS for NP-Hard problems are presented by Hansen, Mladenović, and Moreno Pérez (2010), Polat, Kalayci, Mutlu, and Gupta (2015) and Kalayci, Polat, and Gupta (2016).

Pseudo code for the main steps of the basic VNS algorithm based on Hansen and Mladenović (2001) is given in Fig. 1. First, the set of neighborhood structures ($N_a, a = 1, \ldots, a_{max}$) is selected to use in the search. Then, an initial solution (x^0) is found and a termination condition is chosen. In the shaking step, a point x' is randomly generated from the a^{th} neighborhood of $x(x' \in N_a(x))$ and in the local search step, some local search method is applied to obtain x'' from x'. In the move Or Not step, if the local optimum is better than the incumbent, move ($x \leftarrow x''$) and continue search with $N_1(a \leftarrow 1)$, otherwise set $a \leftarrow a + 1$. These steps are repeated until the termination condition is reached.

3.3. Hybridization of ACS and VNS

ACS is a powerful optimization technique with a distributed long term memory structure while it is desperate for local search in order to obtain efficient results (Dorigo & Gambardella, 1997). Gajpal and Abad (2009) used local search strategies as sub procedures of ant colony system to solve VRPSPD and obtained efficient results. VNS on the other hand, is also another very powerful technique that provides an intensive local search mechanism given that it is used with appropriate neighborhood operators while unfortunately it lacks a memory structure. Polat, Kalayci, Kulak, et al. (2015) recently applied VNS algorithm utilizing a simple perturbation mechanism to VRPSPD with successful results. Table 4 summarizes the highlighted advantages and disadvantages of both algorithms. In this study, the idea is focused on improving the performance of both algorithms to overcome each other's deficiencies when combined together. In the literature, generally, papers that combine ant colony optimization and local search techniques make use of local search operators as a sub procedure of ACS. However, in this study on the opposite, ACS works as a sub procedure of VNS in order to provide a memory mechanism to the integrated algorithm.

4. The proposed solution algorithm

4.1. Solution representation

Designing solution representation scheme is a critical step of metaheuristics since every other step is constructed on top of used representation. A common representation scheme, permutation based representation, is not very useful for our approach since inter-route neighborhood structures that transfer clients from one route to another as described in Section 4.4, are applied. Therefore, a simple, yet very handy matrix representation is used. An example solution representation is presented in Fig. 2 where each number between 1 and 50 represent clients and 0 represents the depot. Fig. 2 also represents the best known solution of CMT6X problem instance mentioned in Section 5.4.2.

4.2. Generating initial solution

The initial solution is constructed by use of the formula presented by Altinel and Oncan (2005) that proposes an enhancement of Clarke and Wright savings algorithm (Clarke & Wright, 1964). See details in the study of Polat, Kalayci, Kulak, et al. (2015).

4.3. Iterative local search with VNS

Following the initial solution generation, a VNS approach is applied to improve the solution over generations by systematically changing the neighborhoods. VNS consists of two main phases, namely shaking and local search, which make use of various neighborhood structures with the intention of exploring the solution space further. The shaking phase, as its name suggests, shakes the current solution with the aim of changing the search direction while local search phase exploits new neighborhoods of the current solution taken from the shaking phase. Section 4.4. presents different neighborhood structures used in both phases. The pseudo code for VNS procedure is given in Fig. 3.

4.4. Neighborhood structures

Intra-route and inter-route neighborhood structures employed in the shaking and local search phases of VNS algorithm is borrowed from Gajpal and Abad (2009), Goksal et al. (2013) and Polat, Kalayci, Kulak, et al. (2015). Note that, similar variants of these structures are commonly used in the literature as mentioned in

Table 4
Main advantages and disadvantages of ACS and VNS.

ACS		VNS	
Advantage: Has a distributed long term memory structure	Disadvantage: Desperate for local search	Advantage: Provides intensive local search	Disadvantage: Lacks memory structure

Route 1:	0	→	27	→	48	8	→	26	→	7	→	43	→	24	→	23	→	6	→	0	→	0
Route 2:	0	→	14	→	25	13	→	41	→	40	→	19	→	42	→	17	→	0	→	0	→	0
Route 3:	0	→	12	→	37	44	→	15	→	45	→	33	→	39	→	10	→	49	→	5	→	0
Route 4:	0	→	46	→	47	4	→	18	→	0	→	0	→	0	→	0	→	0	→	0	→	0
Route 5:	0	→	2	→	20	35	→	36	→	3	→	28	→	31	→	22	→	1	→	0	→	0
Route 6:	0	→	32	→	11	16	→	29	→	21	→	50	→	34	→	30	→	9	→	38	→	0

Fig. 2. An example solution representation for CMT6X problem instance.

```
select N_a, a = 1, ..., a_max as inter-route operators of shaking phase
select N_b, b = 1, ..., b_max as inter-route and intra-route operators of local search phase
find x^0 using saving based heuristic algorithm
Set VNS termination condition
x ← x^0
While VNS termination condition is not met do
    Set a ← 1
    While a < a_max do
        x' ← shake with inter-route structures (N_a(x))
        While b< b_max do
            x'' ← best local search with inter-route and intra-route structures (N_b(x'))
            x ← x''
            a ← 1
            If f(x) < f(x^0) then
                x^0 ← x
                a ← 1
                b ← 1
                break
            Else
                b ← b + 1
            End if
        End while
        a ← a + 1
    End while
End while
```

Fig. 3. Pseudo code for VNS procedure.

Table 5
Inter-route structures employed in shaking step of the algorithm.

Exchange(m, n)	m sequential customers from a randomly selected route (let it be route 1) is transferred to another randomly selected route (let it be route 2) and then n sequential customers from route 2 are transferred to route 1 (Polat, Kalayci, Kulak, et al., 2015).
Cross	Randomly selected two routes are divided into two, namely first part and second part. Then, first part of route 1 is connected to second part of route 2 while the first part of route 2 is connected to the second part of route 1 (Goksal et al., 2013)
Shift(m, 0)	m sequential customers from a randomly selected route is transferred to another randomly selected route.

Section 1. Inter-route structures employed in shaking step of the VNS algorithm are summarized in Table 5, inter-route and intra-route structures employed in the local search phase of the algorithm is summarized in Table 6. The inter-route structures employed in the shaking step applies a random movement between two routes. The inter-route and intra-route structures employed in local search phase of the algorithm, provides the best possible transfer between or within routes.

4.5. Pheromone update rule and perturbation move with ACS

In ACSEVNS algorithm, the best global VNS solution (i.e., the VNS solution which constructed the shortest network from the beginning of the trial) is allowed to deposit pheromone. This rule is performed after VNS improves a solution. The pheromone level is updated by Eq. (2) as presented in Section 3.1. In this study, instead of a simple procedure, ant colony system provides a perturbation move to the integrated algorithm by Eq. (1) as presented in Section 3.1.

4.6. The proposed ant colony system empowered variable neighborhood search algorithm (ACSEVNS)

The pseudo code for ACSEVNS is presented in Fig. 4.

```
Read data for problem instance
Initialize algorithm parameters
select N_a, a = 1, ..., a_max as inter-route operators of shaking phase
select N_b, b = 1, ..., b_max as inter-route and intra-route operators of local search phase
find x^0 using saving based heuristic algorithm
Set VNS termination condition
x ← x^0
While ACOEVNS termination condition is not met do
    Set a ← 1
    While a < a_max do
        x' ← shake with inter-route structures (N_a(x))
        While b< b_max do
            x'' ← best local search with inter-route and intra-route structures (N_b(x'))
            x ← x''
            a ← 1
            If f(x) < f(x^0) then
                x^0 ← x
                a ← 1
                b ← 1
                Release pheromones on the edges using equation (2)
                break
            Else
                b ← b + 1
            End if
        End while
        a ← a + 1
    End while
    If iteration limit for VNS is exceeded
        x ← generate an ant colony solution for perturbation move using equation (1)
    end
End while
```

Fig. 4. Pseudo code for ACOEVNS algorithm.

5. Experimental study

5.1. Implementation

The proposed approach has been modelled and tested using MATLAB R2015b and then recoded in Microsoft Visual C++ 2015 environment for speed purposes. All numerical experiments were run on an Intel Xeon E5-2650 2.0 GHz processor with 32GB RAM. VRPSPD data sets with variety of problem size and characteristics available in the literature are used in order to evaluate the effectiveness of the proposed algorithm. In the following subsections, test problems, parameter settings of the algorithm and computational results for each test problem are presented.

5.2. Test problems

The algorithm is tested on two different group of VRPSPD benchmark data sets which are commonly used for comparison purposes in the literature. The first group that consists of 40 test problems involving 50 customers and a central depot, namely SCA and CON sets, is generated and presented by Dethloff (2001). The second group that consists of 28 test problems involving from 50 to 200 customers and a central depot, namely CMT sets, is generated and presented by Salhi and Nagy (1999). 14 problems of CMT sets do not impose time limit (a limit on the total route length) restrictions while the remaining 14 problems impose time limit restrictions. Regarding CMT sets' solutions, we believe that there has been a confusion on comparing results since some authors used rounded demand configuration while others used unrounded demand configuration for CMT sets in the literature (Zachariadis et al., 2010). There is a further confusion regarding the generation scheme for Y series of CMT sets since some authors (Ai & Kachitvichyanukul, 2009; Avci & Topaloglu, 2015; Çatay, 2010; Goksal et al., 2013; Li et al., 2015; Polat, Kalayci, Kulak, et al. (2015); Subramanian et al., 2010; Subramanian et al., 2011; Tasan & Gen, 2012; Wang, Mu, Zhao, & Sutherland, 2015; Zachariadis & Kiranoudis, 2011; Zachariadis et al., 2009, 2010) used swapping scheme of the delivery and pickup values for every customer while some others (Alfredo Tang Montané & Galvão, 2006; Chen & Wu, 2005; Nagy & Salhi, 2005) used exchanging scheme of the demand and pickup values for every other customer. Therefore, a performance comparison of algorithms applied to such configurations is not possible since such configuration changes completely creates new data and precision of customer demands heavily affects the objective value of the solutions. Therefore, consideration is given to unrounded demand configuration and swapped scheme since they appear to be well accepted approaches in recent years.

5.3. Parameter settings

A robust parameter setting is required for the algorithm to efficiently perform on different data sets. In order to select best pa-

Table 6
Inter-route and intra-route structures employed in local search phase of the algorithm.

Inter-route structures:	
best Exchange(m, n)	m sequential customers from a randomly selected route (let it be route 1) is transferred to best possible location in another route (let it be route 2) and then n sequential customers from route 2 are transferred to the best possible location in route 1 (m, $n \in [0, 3]$) (Gajpal & Abad, 2009)
best Cross	Randomly selected two routes are divided into two, namely first part and second part. Then, first part of route 1 is connected to second part of route 2 while the first part of route 2 is connected to the second part of route 1.
best Shift(m, 0)	m sequential customers from a randomly selected route is transferred to another randomly selected route with the best possible position ($m \in [1, 3]$).
Intra-route structures:	
best swap	The best permutation movement that swaps the locations of two clients in the same route is applied to the routes that were previously shaken in the shaking phase. In other words, the best possible swap is employed within the routes.
best insert	The best insertion movement that removes a client from its position and inserts to another position in the same route is applied to the routes that were previously shaken in the shaking phase.
best edge insert	The best edge insertion movement that removes two sequential clients from their position and inserts to another position in the same route is applied to the routes previously shaken in the shaking phase.
best two opt	The best two-opt movement that swaps pairs of links in the same route is applied to the routes previously shaken in the shaking phase.

Table 7
Parameter settings summary of the proposed algorithm.

Parameter	Levels tested	Value selected
q_0	$\{0.5, 0.7, 0.9\}$	0.9
β	$\{0.5, 2, 5\}$	5
ρ	$\{0.05, 0.1, 0.4\}$	0.05
ilimit	$\{n/2, n, 2n\}$	n

rameter setting, tests were performed on four parameters: q_0 (ant colony rule application parameter), β (a parameter which determines the relative importance of pheromone vs. heuristic information), ρ (pheromone decay parameter) and *ilimit* (number of iterations to make VNS wait until ACS procedure is summoned for perturbation move). Since no obvious correlation between the optimal settings of these parameters was observed, each one of them is tested individually for deciding on the standard parameter setting. Table 7 summarizes the parameter settings for the proposed algorithm. Taking q_0 value as 0.9 yielded the best results. Note that, fixing q_0 value to 1, forces the ACS to exploit only while a small ratio of biased exploration was helpful. Regarding β parameter, taking value as 5 made the algorithm produce slightly faster results, yet not caused any significant change on the quality of the solutions. Taking ρ parameter very small lets the pheromone levels decrease slowly throughout generations of the algorithm and performed better results. *ilimit* parameter, which decides when ACS is called, was tested with the following levels: $\{n/2, n, 2n\}$. The first level turned out to be faster, yet less effective while the last level performed better, yet performed slower. Therefore, the medium level of *ilimit* had the best performance overall with a balance of CPU time and efficiency.

5.4. Computational results

The following subsections presents the computational results of the proposed algorithm for the VRPSPD.

5.4.1. Results of ACSEVNS for Dethloff's (2001) data set

To the best of our knowledge, the best known upper bounds for Dethloff's (2001) data set were found by the following algorithms:

ACS: Ant Colony System (Gajpal & Abad, 2009)
PILS: Parallel iterated local search (A. Subramanian et al., 2010)
VLBR: Variable Length Bone Route (Zachariadis et al., 2010)
HPSO: Hybrid particle swarm optimization (Goksal et al., 2013)

Therefore, we compare the performance of the proposed approach against these algorithms reported in the literature. Table 8 presents the computational results of compared algorithms, ACS (Gajpal & Abad, 2009) and ACSEVNS over 40 instances. Bold numbers given in the table indicate that the algorithms have reached the best solution. Table 8 also reports average gap to the best known solution (BKS) as percent, number of best known solutions found, average computation time over 40 instances in last rows. The average percentage gap of the solutions of ACSEVNS from the best solutions are 0.00%. As presented in Table 8, the average computational time for the ACSEVNS approach is approximately 6 s for the benchmark problem instances of Dethloff (2001). It is worth to note that the results reported in Table 6 results are proved to be optimal by Subramanian et al. (2011). Therefore, further improvements on solution quality are not possible.

5.4.2. Results of ACOEVNS for Salhi and Nagy's (1999) data set

In this subsection, firstly, the performance of ACOEVNS algorithm is compared to ACS (Gajpal & Abad, 2009) and PVNS (Polat, Kalayci, Kulak, et al., 2015) on Salhi and Nagy's (1999) data set since the proposed ACOEVNS algorithm is built up on these two algorithms.

It is not straightforward to compare CPU times since different algorithms are tested on different computers. In order to give an idea regarding the CPU performance of each machine used to implement, Table 9 presents CPU score according to PassMark benchmark software where higher scores represent better performances. However, it is also known that programming languages perform differently for different algorithms. Moreover, programming skills has a significant effect on how fast an algorithm converges to optimal solutions. Therefore, the consideration is given to the quality of solutions rather than CPU time as long as the solution time is within acceptable range for a metaheuristic algorithm.

As presented in Table 10, the average computational time for the ACSEVNS approach is approximately 178 seconds for the benchmark problem instances of Salhi and Nagy (1999). Tables 10 reports that the ACSEVNS algorithm produces efficient solutions in reasonable time for the benchmark problems of Salhi and Nagy (1999). Another conclusion that can be drawn from Table 10 is that the idea of combining ACS and VNS algorithms works since ACSEVNS outperformed ACS and PVNS.

To the best of our knowledge, the best known upper bounds for Salhi and Nagy's (1999) data set with rounded demand configuration were found by the following algorithms:

LNS: Large Neighborhood Search (Ropke & Pisinger, 2006)

Table 8
Computational results for Dethloff's (2001) data set.

Instance	Best so far known solution in the literature		ACS (Gajpal & Abad, 2009)				ACSEVNS			
	Ref.	BKS	best	Single run	t	gap%	best	avg	t	gap%
SCA3-0	PILS, VLBR, HPSO	**635.62**	**635.62**	635.62	6.00	0.00	**635.62**	635.62	4.77	0.00
SCA3-1	PILS, VLBR, HPSO	**697.84**	**697.84**	697.84	6.00	0.00	**697.84**	697.84	5.24	0.00
SCA3-2	PILS, VLBR, HPSO	**659.34**	**659.34**	659.34	6.00	0.00	**659.34**	659.34	7.47	0.00
SCA3-3	PILS, VLBR, HPSO	**680.04**	**680.04**	680.04	6.10	0.00	**680.04**	680.34	5.20	0.00
SCA3-4	PILS, VLBR, HPSO	**690.50**	**690.50**	690.50	5.70	0.00	**690.50**	690.50	4.96	0.00
SCA3-5	PILS, VLBR, HPSO	**659.91**	**659.91**	659.91	5.10	0.00	**659.91**	659.91	5.18	0.00
SCA3-6	PILS, VLBR, HPSO	**651.09**	**651.09**	651.09	6.10	0.00	**651.09**	651.11	4.68	0.00
SCA3-7	PILS, VLBR, HPSO	**659.17**	**659.17**	659.17	6.80	0.00	**659.17**	659.17	6.06	0.00
SCA3-8	PILS, VLBR, HPSO	**719.48**	**719.47**	719.47	5.40	0.00	**719.48**	719.56	4.51	0.00
SCA3-9	PILS, VLBR, HPSO	**681.00**	**681.00**	681.00	6.00	0.00	**681.00**	681.00	7.08	0.00
SCA8-0	PILS, VLBR, HPSO	**961.50**	**961.50**	961.50	11.00	0.00	**961.50**	961.50	5.33	0.00
SCA8-1	PILS, VLBR, HPSO	**1049.65**	**1049.65**	1050.38	11.50	0.00	**1049.65**	1049.65	5.62	0.00
SCA8-2	PILS, VLBR, HPSO	**1039.64**	**1042.69**	1044.48	11.90	0.00	**1039.64**	1041.62	6.05	0.00
SCA8-3	PILS, VLBR, HPSO	**983.34**	**983.34**	983.34	11.30	0.00	**983.34**	983.34	8.39	0.00
SCA8-4	PILS, VLBR, HPSO	**1065.49**	**1065.49**	1065.49	11.10	0.00	**1065.49**	1065.49	6.07	0.00
SCA8-5	PILS, VLBR, HPSO	**1027.08**	**1027.08**	1027.08	11.30	0.00	**1027.08**	1027.14	6.96	0.00
SCA8-6	PILS, VLBR, HPSO	**971.82**	**971.82**	971.82	12.00	0.00	**971.82**	971.82	7.76	0.00
SCA8-7	PILS, VLBR, HPSO	**1051.28**	**1052.17**	1063.15	12.50	0.00	**1051.28**	1051.28	8.14	0.00
SCA8-8	PILS, VLBR, HPSO	**1071.18**	**1071.18**	1071.18	11.00	0.00	**1071.18**	1071.22	7.06	0.00
SCA8-9	PILS, VLBR, HPSO	**1060.50**	**1060.50**	1061.23	11.50	0.00	**1060.50**	1060.50	5.29	0.00
CON3-0	PILS, VLBR, HPSO	**616.52**	**616.52**	616.52	8.30	0.00	**616.52**	616.52	6.80	0.00
CON3-1	PILS, VLBR, HPSO	**554.47**	**554.47**	554.47	7.10	0.00	**554.47**	554.47	5.01	0.00
CON3-2	PILS, VLBR, HPSO	**518.00**	**518.00**	519.11	6.90	0.00	**518.00**	518.00	7.55	0.00
CON3-3	PILS, VLBR, HPSO	**591.19**	**591.19**	591.19	7.20	0.00	**591.19**	591.19	5.75	0.00
CON3-4	PILS, VLBR, HPSO	**588.79**	**588.79**	588.79	6.00	0.00	**588.79**	588.79	3.90	0.00
CON3-5	PILS, VLBR, HPSO	**563.70**	**563.70**	563.70	6.90	0.00	**563.70**	563.70	6.86	0.00
CON3-6	PILS, VLBR, HPSO	**499.05**	**499.05**	499.05	7.30	0.00	**499.05**	499.05	8.54	0.00
CON3-7	PILS, VLBR, HPSO	**576.48**	**576.48**	576.48	7.00	0.00	**576.48**	576.48	4.26	0.00
CON3-8	PILS, VLBR, HPSO	**523.05**	**523.05**	523.05	7.40	0.00	**523.05**	523.05	3.89	0.00
CON3-9	PILS, VLBR, HPSO	**578.25**	**578.25**	578.25	6.80	0.00	**578.25**	578.25	6.33	0.00
CON8-0	PILS, VLBR, HPSO	**857.17**	**857.17**	857.17	12.30	0.00	**857.17**	857.17	5.40	0.00
CON8-1	PILS, VLBR, HPSO	**740.85**	**740.85**	740.85	12.00	0.00	**740.85**	740.85	8.46	0.00
CON8-2	PILS, VLBR, HPSO	**712.89**	**712.89**	712.89	13.00	0.00	**712.89**	712.89	4.79	0.00
CON8-3	PILS, VLBR, HPSO	**811.07**	**811.07**	811.07	13.90	0.00	**811.07**	811.07	7.21	0.00
CON8-4	PILS, VLBR, HPSO	**772.25**	**772.25**	772.25	11.90	0.00	**772.25**	772.25	6.70	0.00
CON8-5	PILS, VLBR, HPSO	**754.88**	**754.88**	754.88	12.40	0.00	**754.88**	754.88	5.74	0.00
CON8-6	PILS, VLBR, HPSO	**678.92**	**678.92**	678.92	12.40	0.00	**678.92**	678.92	4.36	0.00
CON8-7	PILS, VLBR, HPSO	**811.96**	**811.96**	811.96	13.00	0.00	**811.96**	813.46	8.38	0.00
CON8-8	PILS, VLBR, HPSO	**767.53**	**767.53**	767.53	12.50	0.00	**767.53**	767.53	6.16	0.00
CON8-9	PILS, VLBR, HPSO	**809.00**	**809.00**	809	12.90	0.00	**809.00**	809.00	7.19	0.00
G. Avg.					9.29	0.00			6.13	0.00
BKS found		40	40				40			

The best solution of each problem instance is highlighted in bold.
BKS: The best known solution; **Ref.:** The best solution reference; **best:** The best solution in 10 replications; **avg.:** Average solution over 10 replications; **Single run:** Single run solution of ACS; **t:** Average computation time; **gap%:** Percentage difference between the best known and the best found solution; **G. Avg.:** Average of 40 instances; **BKS found**: number of best solutions found

Table 9
CPU performance comparison.

Algorithm	CPU used to implement	Programming Language	Score[a]	Ratio[b]
PVNS (Polat, Kalayci, Kulak, et al., 2015)	Intel Core 2 Duo 2.0 GHz	C#	1091	4.68
ACS (Gajpal & Abad, 2009)	Intel Xeon 2.4 GHz	C	276	1.18
ACSEVNS	Intel Xeon 2.0 GHz	C++	233	1

[a] CPU Mark Rating score as of 29th of December 2015 - Higher results represent better performance.
[b] scaled CPU performance.

PSO: Particle Swarm Optimization (Ai & Kachitvichyanukul, 2009)
TSGLS: Tabu Search and Guided Local Search (Zachariadis et al., 2009)
ACS: Ant Colony System (Gajpal & Abad, 2009)
SBAA: Saving Based Ant Algorithm (Çatay, 2010)
VLBR: Variable Length Bone Route (Zachariadis et al., 2010)
PILS: Parallel Iterated Local Search (A. Subramanian et al., 2010)
ILS: Iterated Local Search (Jun & Kim, 2012)
HPSO: Hybrid Particle Swarm Optimization (Goksal et al., 2013)
PVNS: Perturbation based Variable Neighborhood Search (Polat, Kalayci, Kulak, et al., 2015)
ILSANS: Iterated Local Search embedded Adaptive Neighborhood Selection (Li et al., 2015)

Therefore, a comparison of computational results with the best known solutions for Salhi and Nagy's (1999) data set is provided in Table 10 where found solutions is matched with the related reference in the literature.

Table 10
Comparison of ACSEVNS with ACS and PVNS on Salhi and Nagy's (1999) data set.

Instance			ACSEVNS			ACS (Gajpal & Abad, 2009)				PVNS (Polat, Kalayci, Kulak, et al., 2015))			
CMT	#N	#v	best	avg.	t	best	Single run.	t	gap%	best	avg.	t	gap%
1X	50	3	**466.77**	466.77	8.50	**466.77**	466.77	5.00	0.00	**466.77**	466.77	16.52	0.00
1Y	50	3	**466.77**	466.77	8.50	**466.77**	466.77	5.00	0.00	**466.77**	466.77	8.26	0.00
2X	75	6	**684.21**	684.21	32.50	**684.21**	688.05	20.75	0.00	**684.21**	684.29	44.92	0.00
2Y	75	6	**684.21**	684.21	36.50	684.94	688.26	22.25	0.11	**684.21**	684.21	46.73	0.00
3X	100	4	**721.27**	721.27	45.20	721.40	721.4	41.25	0.02	**721.27**	721.30	52.18	0.00
3Y	100	4	**721.27**	721.27	40.30	721.40	724.54	43.75	0.02	**721.27**	721.27	46.09	0.00
4X	150	7	**852.46**	852.46	142.10	854.12	857.19	131.75	0.19	**852.46**	852.48	118.97	0.00
4Y	150	7	**852.46**	852.46	136.35	855.76	860.85	140.25	0.39	**852.46**	852.48	136.37	0.00
5X	199	10	**1030.55**	1030.57	420.15	1034.87	1035.03	377.50	0.42	**1030.55**	1030.62	554.39	0.00
5Y	199	10	**1030.55**	1030.62	410.50	1037.34	1039.99	393.50	0.66	**1030.55**	1030.66	287.05	0.00
6X	50	6	**555.43**	555.43	32.50	**555.43**	555.43	14.00	0.00	**555.43**	555.43	47.00	0.00
6Y	50	6	**555.43**	555.43	32.30	**555.43**	555.43	13.75	0.00	**555.43**	555.43	47.30	0.00
7X	75	11	**900.12**	900.12	52.55	**900.12**	901.11	47.75	0.00	901.22	901.22	70.30	0.12
7Y	75	11	**900.12**	900.12	56.30	900.54	901.22	46.25	0.05	901.22	901.22	69.80	0.12
8X	100	9	**865.50**	865.50	120.25	**865.50**	865.5	80.75	0.00	**865.50**	865.50	224.60	0.00
8Y	100	9	**865.50**	865.50	127.50	**865.50**	865.5	77.75	0.00	**865.50**	865.50	162.70	0.00
9X	150	14	**1160.68**	1160.96	360.20	1161.54	1161.97	300.00	0.07	1161.24	1161.72	483.40	0.05
9Y	150	14	**1160.68**	1160.92	350.80	1161.54	1161.97	291.75	0.07	1161.24	1161.66	477.10	0.05
10X	200	18	**1373.40**	1375.77	880.50	1386.29	1401.13	773.50	0.93	1388.25	1388.72	1168.80	1.07
10Y	200	18	**1373.40**	1375.77	860.25	1395.04	1400.68	757.50	1.56	1388.25	1388.77	1112.10	1.07
11X	120	4	**833.92**	833.92	42.45	839.66	844.52	57.25	0.69	**833.92**	834.05	33.91	0.00
11Y	120	4	**833.92**	833.92	40.50	840.19	859.57	52.75	0.75	**833.92**	834.00	49.64	0.00
12X	100	5	**662.22**	662.22	38.25	663.01	663.09	36.25	0.12	**662.22**	662.22	33.91	0.00
12Y	100	5	**662.22**	662.22	41.50	663.50	663.5	39.25	0.19	**662.22**	662.26	33.34	0.00
13X	120	11	**1542.86**	1542.86	260.25	**1542.86**	1542.86	160.25	0.00	**1542.86**	1543.17	332.70	0.00
13Y	120	11	**1542.86**	1542.86	250.30	**1542.86**	1542.86	160.25	0.00	**1542.86**	1542.86	375.30	0.00
14X	100	10	**821.75**	821.75	85.50	**821.75**	821.75	78.50	0.00	**821.75**	821.75	228.50	0.00
14Y	100	10	**821.75**	821.75	83.70	**821.75**	821.75	74.75	0.00	**821.75**	821.75	204.60	0.00
G. Avg.					178.44			151.54	**0.22**			230.95	**0.09**

The best solutions of each problem instance is highlighted in bold.
#N: The number of clients; **#v:** The number of routes in the best known solution; **BKS:** The best known solution; **Ref.:** The best solution reference; **best:** The best solution in 10 replications; **avg.:** Average solution over 10 replications; **t:** Average computation time; **gap%:** Percentage difference between the best known and the best found solution; **G.Avg.:** Average of 28 instances.

Table 10
Computational results for Salhi and Nagy's (1999) data set.

Instance			Best so far known solution in the literature			ACSEVNS		
CMT	#N	#v	BKS	Ref.	#v	best	avg.	gap%
1X	50	3	**466.77**	LNS, PSO, HPSO, ACS, PILS, ILS, PVNS, ILSANS	3	**466.77**	466.77	0.00
1Y	50	3	**466.77**	LNS, PSO, HPSO, ACS, PILS, PVNS, ILSANS	3	**466.77**	466.77	0.00
2X	75	6	**684.21**	ACS, TSGLS, VLBR, PILS, HPSO, PVNS	6	**684.21**	684.21	0.00
2Y	75	6	**684.21**	ACS, TSGLS, VLBR, PILS, HPSO, PVNS	6	**684.21**	684.21	0.00
3X	100	4	**721.27**	TSGLS, PILS, VLBR, HPSO, PVNS	4	**721.27**	721.27	0.00
3Y	100	4	**721.27**	TSGLS, PILS, VLBR, HPSO, PVNS	4	**721.27**	721.27	0.00
4X	150	7	**852.46**	TSGLS, PILS, VLBR, HPSO, PVNS, ILSANS	7	**852.46**	852.46	0.00
4Y	150	7	**852.46**	TSGLS, PILS, VLBR, HPSO, PVNS	7	**852.46**	852.46	0.00
5X	199	9	**1029.25**	PILS	10	1030.55	1030.57	0.10
5Y	199	9	**1029.25**	PILS	10	1030.55	1030.62	0.10
6X	50	6	**555.43**	ACS, PVNS, ILSANS	6	**555.43**	555.43	0.00
6Y	50	6	**555.43**	ACS, PVNS, ILSANS	6	**555.43**	555.43	0.00
7X	75	11	**900.12**	ACS	11	**900.12**	900.12	0.00
7Y	75	11	900.54	ACS	11	**900.12**	900.12	**−0.05**
8X	100	9	**865.50**	LNS, ILS, ACS, SBAA, PVNS, ILSANS	9	**865.50**	865.50	0.00
8Y	100	9	**865.50**	ILS, ACS, SBAA, PVNS, ILSANS	9	**865.50**	865.50	0.00
9X	150	14	1161.24	PVNS	14	**1160.68**	1160.96	**−0.05**
9Y	150	14	1161.24	PVNS	14	**1160.68**	1160.92	**−0.05**
10X	200	18	1386.29	ACS	18	**1373.40**	1375.77	**−0.93**
10Y	200	18	1388.25	PVNS	18	**1373.40**	1375.77	**−1.07**
11X	120	4	**833.92**	PILS, VLBR, HPSO, PVNS	4	**833.92**	833.92	0.00
11Y	120	4	**833.92**	PILS, VLBR, HPSO, PVNS	4	**833.92**	833.92	0.00
12X	100	5	**662.22**	TSGLS, PILS, VLBR, HPSO, PVNS	5	**662.22**	662.22	0.00
12Y	100	5	**662.22**	TSGLS, PILS, VLBR, HPSO, PVNS	5	**662.22**	662.22	0.00
13X	120	11	**1542.86**	ACS,PVNS	11	**1542.86**	1542.86	0.00
13Y	120	11	**1542.86**	ACS,PVNS	11	**1542.86**	1542.86	0.00
14X	100	10	**821.75**	ILS, ACS, SBAA, PVNS, ILSANS	10	**821.75**	821.75	0.00
14Y	100	10	**821.75**	ILS, ACS, PVNS, ILSANS	10	**821.75**	821.75	0.00
G. Avg.								**−0.07**

The best solution of each problem instance is highlighted in bold.
#N: The number of clients; **#v:** The number of routes in the best known solution; **BKS:** The best known solution; **Ref.:** The best solution reference; **best:** The best solution in 10 replications; **avg.:** Average solution over 10 replications; **t:** Average computation time; **gap%:** Percentage difference between the best known and the best found solution; **G.Avg.:** Average of 28 instances

Table 11
Strengths and weaknesses of the proposed solution approach.

Strengths	Weaknesses
Has direct applications in people and goods transportation	Requires relatively more of parameters
Flexible	Sensitive to parameters
Simple	
Robust	
Parallelizable	

Table 10 confirms the effectiveness of the algorithm as it was able to catch almost all of the best known solutions in the literature except CMT5X and CMT5Y. Moreover, improvements were achieved on the following problem instances: CMT7Y, CMT9X, CMT9Y, CMT10X and CMT10Y.

6. Discussion

The proposed hybrid algorithm that effectively solves VRPSPD has direct applications in people and goods transportation. Even though the proposed hybrid algorithm is tested only on simultaneous pickup and delivery cases, it can be extended to solve many other VRP variants with little modifications. Therefore, it is highly flexible which is important for software developers designing vehicle routing solvers for several users who are in need of solving different types of VRP. Moreover, this algorithm is easy to understand and implement since diversification and intensification phases are separately tackled by two nested algorithms. Therefore, it satisfies the simplicity feature that is highly desirable for a metaheuristic algorithm. Since the proposed algorithm is very successful on consistently returning good solutions for many benchmark problems, it is highly robust that can be concluded by inspecting the detailed results of tests presented in previous sections. Although the speed of a single processor has notably improved in the last three decades, its computational capability may be inadequate in solving complex and bigger sized problems fast enough. Since processors have also been equipped with many cores that enables parallel processing capability, it is advisable to parallelize algorithms in order to take full advantage of next generation hardware. Although in this study, the proposed algorithm is not configured to work in parallel, it is highly parallelizable simply by launching several copies on many cores starting from a different initial solution and cooperating with each other. Thus, good solutions can be obtained even faster where time greatly matters. Ideally, metaheuristics that have few control parameters and less sensitive to the parameters is desired. Although VNS requires less parameters, ACS requires relatively more parameters. Consequently, the combined algorithm can be considered to be weak in terms of number of parameters and their sensitivity to level combinations. Table 11 summarizes the strengths and weaknesses of proposed solution approach mentioned above.

Theoretically, the proposed solution approach that combines ACS and VNS has an intriguing idea since on the contrary of the literature, ACS works as a sub-procedure of the main VNS algorithm to provide a perturbation mechanism. Thus, an alternative diversification strategy is introduced by taking advantage of the global search capability of ACS.

The findings provide a few more insightful implications. Firstly, it contributes to the vehicle routing literature by proposing an effective solution approach that provided new best solutions as a reference for comparison purposes of future studies in VRPSPD literature. Moreover, the approach can be easily adapted to solve other VRP variants as well as other combinatorial optimization problems in the class of NP-Hard problems since the balance of intensification and diversification is always desired for a metaheuristic algorithm. Finally, this novel idea can trigger the design of new algorithms as it provides a spotlight on the importance of diversification and intensification strategies. As better optimization yielding to better results in a shorter time will be crucial for expert and intelligent systems in the near future, any improvements on algorithms' performance in terms of robustness, simplicity, flexibility or speed are greatly valued.

7. Concluding remarks

Combination of powerful computers and high performance algorithms integrated to expert and intelligent software systems will be crucial in order to race against time in any field of the near future. In this study, we proposed a hybrid metaheuristic algorithm that combines the ant colony system and the variable neighborhood search method to solve the vehicle routing problem with simultaneous pick-up and delivery with and without time limit restrictions. Ant colony system (ACS) algorithm's advantage lies in the fact that it utilizes a distributed long term memory structure. A disadvantage of ACS would be the need of utilizing local search strategies in order to obtain efficient results. Variable neighborhood search (VNS) algorithm on the other hand, has an advantage of providing intensive local search while it lacks a memory structure. Disadvantages of ACS and VNS are widely reported in the literature. In the proposed ant colony system empowered variable neighborhood search algorithm (ACSEVNS), the idea is focused on improving the performance of both algorithms to overcome each other's deficiencies when combined together. In this approach, variable neighborhood search approach released pheromones on the edges while ants selected paths using this pheromone information and thus ant colony system provided a perturbation mechanism to guide the variable neighborhood search algorithm jump from local optimum solutions. In this way, VNS utilizes a memory mechanism by the help of ACS. Computational experiment with benchmark problem instances has shown that in general the proposed ant colony system empowered variable neighborhood search (ACSEVNS) algorithm outperformed VNS with simple perturbation mechanism and ACS with embedded local search strategies. Furthermore, the proposed algorithm performed well in terms of the solution quality and the CPU time compared to the algorithms reported in the literature. Further research can be done to apply this strategy to other vehicle routing problem variants. Moreover, it is worth to try parallelization strategies utilizing different threads of CPUs for ACSEVNS that can improve the performance of the algorithm.

Acknowledgment

This research is funded by the Scientific Research Project Coordination Unit of Pamukkale University (PAUBAP) with the grant number 2015FBE023.

References

Ai, T. J., & Kachitvichyanukul, V. (2009). A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research, 36*, 1693–1702.

Alfredo Tang Montané, F., & Galvão, R. D. (2006). A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers & Operations Research, 33*, 595–619.

Altinel, I. K., & Oncan, T. (2005). A new enhancement of the Clarke and Wright savings heuristic for the capacitated vehicle routing problem. *Journal of the Operational Research Society, 56*, 954–961.

Avci, M., & Topaloglu, S. (2015). An adaptive local search algorithm for vehicle routing problem with simultaneous and mixed pickups and deliveries. *Computers & Industrial Engineering, 83*, 15–29.

Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., & Laporte, G. (2007). Static pickup and delivery problems: A classification scheme and survey. *TOP, 15*, 1–31.

Bianchessi, N., & Righini, G. (2007). Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. *Computers & Operations Research, 34*, 578–594.

Chen, J. F., & Wu, T. H. (2005). Vehicle routing problem with simultaneous deliveries and pickups. *Journal of the Operational Research Society*, *57*, 579–587.

Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research, 12*, 568–581.

Coelho, I. M., Munhoz, P. L. A., Ochi, L. S., Souza, M. J. F., Bentes, C., & Farias, R. (2015). An integrated CPU–GPU heuristic inspired on variable neighbourhood search for the single vehicle routing problem with deliveries and selective pickups. *International Journal of Production Research*, 1–18.

Crispim, J., & Brandao, J. (2005). Metaheuristics applied to mixed and simultaneous extensions of vehicle routing problems with backhauls. *Journal of the Operational Research Society, 56*, 1296–1302.

Çatay, B. (2010). A new saving-based ant algorithm for the vehicle routing problem with simultaneous pickup and delivery. *Expert Systems with Applications, 37*, 6809–6817.

Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management Science, 6*, 80–91.

Dell'Amico, M., Righini, G., & Salani, M. (2006). A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. *Transportation Science, 40*, 235–247.

Dethloff, J. (2001). Vehicle routing and reverse logistics: The vehicle routing problem with simultaneous delivery and pick-up. *OR-Spektrum, 23*, 79–96.

Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation, 1*, 53–66.

Gajpal, Y., & Abad, P. (2009). An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup. *Computers & Operations Research, 36*, 3215–3223.

Goksal, F. P., Karaoglan, I., & Altiparmak, F. (2013). A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery. *Computers & Industrial Engineering, 65*, 39–53.

Hansen, P., & Mladenović, N. (2001). Variable neighborhood search: Principles and applications. *European Journal of Operational Research, 130*, 449–467.

Hansen, P., Mladenović, N., & Moreno Pérez, J. (2010). Variable neighbourhood search: Methods and applications. *Annals of Operations Research, 175*, 367–407.

Jun, Y., & Kim, B.-I. (2012). New best solutions to VRPSPD benchmark problems by a perturbation based algorithm. *Expert Systems with Applications, 39*, 5641–5648.

Kalayci, C. B., Polat, O., & Gupta, S. M. (2016). A hybrid genetic algorithm for sequence-dependent disassembly line balancing problem. *Annals of Operations Research, 242*, 321–354.

Li, J., Pardalos, P. M., Sun, H., Pei, J., & Zhang, Y. (2015). Iterated local search embedded adaptive neighborhood selection approach for the multi-depot vehicle routing problem with simultaneous deliveries and pickups. *Expert Systems with Applications, 42*, 3551–3561.

Min, H. (1989). The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A: General, 23*, 377–386.

Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research, 24*, 1097–1100.

Nagy, G., & Salhi, S. d. (2005). Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European Journal of Operational Research, 162*, 126–141.

Polat, O., Gunther, H.-O., & Kulak, O. (2014). The feeder network design problem: Application to container services in the Black Sea region. *Maritime Econ Logistics, 16*, 343–369.

Polat, O., Kalayci, C. B., Kulak, O., & Günther, H.-O. (2015). A perturbation based variable neighborhood search heuristic for solving the vehicle routing problem with simultaneous pickup and delivery with time limit. *European Journal of Operational Research, 242*, 369–382.

Polat, O., Kalayci, C. B., Mutlu, Ö., & Gupta, S. M. (2015). A two-phase variable neighbourhood search algorithm for assembly line worker assignment and balancing problem type-II: An industrial case study. *International Journal of Production Research, 54*, 722–741.

Polat, O., Kulak, O., & Günther, H.-O. (2012). An adaptive neighborhood search approach for VRPSPDTL. In *The second international conference on logistics and maritime systems* Bremen, Germany, (pp. 429–437).

Ropke, S., & Pisinger, D. (2006). A unified heuristic for a large class of Vehicle Routing Problems with Backhauls. *European Journal of Operational Research, 171*, 750–775.

Salhi, S., & Nagy, G. (1999). A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. *The Journal of the Operational Research Society, 50*, 1034–1042.

Subramanian, A., Drummond, L. M. A., Bentes, C., Ochi, L. S., & Farias, R. (2010). A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research, 37*, 1899–1911.

Subramanian, A., Uchoa, E., Pessoa, A., & Ochi, L. (2013). Branch-cut-and-price for the vehicle routing problem with simultaneous pickup and delivery. *Optimization Letters, 7*, 1569–1581.

Subramanian, A., Uchoa, E., Pessoa, A. A., & Ochi, L. S. (2011). Branch-and-cut with lazy separation for the vehicle routing problem with simultaneous pickup and delivery. *Operations Research Letters, 39*, 338–341.

Tasan, A. S., & Gen, M. (2012). A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries. *Computers & Industrial Engineering, 62*, 755–761.

Tovey, C. A. (2002). Tutorial on Computational Complexity. *Interfaces, 32*, 30–61.

Wang, C., Mu, D., Zhao, F., & Sutherland, J. W. (2015). A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup–delivery and time windows. *Computers & Industrial Engineering, 83*, 111–122.

Wassan, N., Wassan, A. H., & Nagy, G. (2008). A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries. *Journal of Combinatorial Optimization, 15*, 368–386.

Wei, L., Zhang, Z., Zhang, D., & Lim, A. (2015). A variable neighborhood search for the capacitated vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research, 243*, 798–814.

Zachariadis, E. E., & Kiranoudis, C. T. (2011). A local search metaheuristic algorithm for the vehicle routing problem with simultaneous pick-ups and deliveries. *Expert Systems with Applications, 38*, 2717–2726.

Zachariadis, E. E., Tarantilis, C. D., & Kiranoudis, C. T. (2009). A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service. *Expert Systems with Applications, 36*, 1070–1081.

Zachariadis, E. E., Tarantilis, C. D., & Kiranoudis, C. T. (2010). An adaptive memory methodology for the vehicle routing problem with simultaneous pick-ups and deliveries. *European Journal of Operational Research, 202*, 401–411.