ACO–Based Neighborhoods for Fixed–charge Capacitated Multi–commodity Network Design Problem

Masoud Yaghini[1], Amir Foroughi[2]

Received: 07. 06. 2013 Accepted: 22.02.2014

Abstract

The fixed-charge Capacitated Multi-commodity Network Design (CMND) is a well-known problem of both practical and theoretical significance. Network design models represent a wide variety of planning and operation management issues in transportation telecommunication, logistics, production and distribution. In this paper, Ant Colony Optimization (ACO) based neighborhoods are proposed for CMND problem. In the proposed neighborhoods, first, an open arc based on the incumbent solution is closed; then, by using an ant colony optimization algorithm called Ant Colony System (ACS), a new solution is generated by constructing new paths for the demands delivered on the closed arc. An algorithm is presented to construct new paths by using ACS algorithm for demands with continuous volume. A sub mixed integer programming (MIP) model is then created by joining the ACS and incumbent solutions. The generated sub-MIP is solved by using an MIP solver and its solution is considered as a neighborhood. In order to evaluate the proposed neighborhoods, an algorithm is developed. The algorithm parameters are tuned by using design of experiments. To assess the algorithm, several benchmark problems with different sizes are used. The statistical analysis shows the efficiency and effectiveness of the proposed algorithm compared to the best approaches found in the literature.

Keywords: Ant Colony Optimization(ACO), ACO-Based neighborhoods, Fixed-charge capacitated multi-commodity network design, meta-heuristic.

Corresponding author E-mail: yaghini@iust.ac.ir

1. Assistant Professor, Department of Rail Transportation Engineering, Iran University of Science and Technology, Tehran, Iran.

2. MSc Grad., Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran.

1. Introduction

Network design models have some applications in a wide variety of planning and operation management issues in telecommunication, logistics, distribution, and transportation specifically railways transportation [Balakrishnan et al., 1997; Magnanti and Wong, 1986; Minoux, 1986; Crainic, 1999; Yaghini and Kazemzadeh, 2012a; Yaghini et al., 2012b; Yaghini et al., 2013a].

The goal of fixed-charge Capacitated Multi-commodity Network Design (CMND) problem is to find the optimal design network on a given directed graph, while selecting arcs to connect a set of nodes and satisfying demands by determining the amount of flow on each arc.

The CMND formulation is modeled as a Mixed Integer Programming (MIP) where continuous variables represent the amount of flows and binary variables determine which arcs are to be used. This problem is categorized as NP-hard [Magnanti and Wong, 1986]. Several formulations for CMND are presented by [Gendron and Crainic, 1994] and their corresponding linear relaxations are compared. The conclusion proves that simplex-based branch-and-bound methods providing the lower bound by the linear programming relaxations, are unlikely to solve even small instances. Although some exact methods are developed to solve CMND problems [Crainic et al. 2001; Holmberg and Yuan, 2000], they have proved to be ineffective on reasonably large problems with many commodities.

Moreover, many heuristic and metaheuristic approaches which were presented to solve CMND are not entirely satisfactory, because the search space does not go beyond local exploration. In fact, they clearly state that their goal was to only explore the behavior of the neighborhood, and that more refined search methods would have to be developed. Indeed, Lagrangian relaxation [Crainic et al. 2001] and cutting planes generation [Chouman et al., 2003; Costa, 2005; Chouman et al. 2009] are used in most of the literature in exact approaches to CMND.

Several heuristic methods have also been proposed for CMND. A combination algorithm of a Lagrangian relaxation method and a branch-and-bound algorithm for reaching solution is proposed in [Holmberg and Yuan, 2000]. A resource decomposition heuristic based on a resource-directive decomposition algorithm has been developed by [Gendron and Crainic, 1994; Gendron and Crainic, 1996]. A tabu search based on a path formulation is described by [Crainic et al., 2000], where the neighborhoods are defined by simplex pivots, and column generation approach dynamically adds new paths to the formulation. A different tabu search heuristic based on an arc formulation is presented by [Ghamlouche et al. 2003], where moving flows around cycles obtains the neighborhoods. This approach is later improved in [Ghamlouche et al., 2004] by adding a path relinking search. Another scatter search algorithm is proposed by [Álvarez et al., 2005]. [Crainic and Gendreau, 2002] proposed a cooperative parallel tabu search. A slope scaling heuristic is proposed by [Crainic et al., 2004] where a Lagrangian perturbation scheme with intensification and diversification mechanisms based on long-term memory are combined. A multilevel cooperative parallel tabu search for CMND is described by [Crainic et al., 2006]. It is based on the principles of local interactions among cooperating searches and controlled diffusion of information within the framework of a multilevel algorithm. [Martín and González, 2010] presented a heuristic method combining mathematical programming techniques called local branching. The method uses an MIP solver to search solution neighborhoods defined by introducing linear inequalities in a mathematical model of the problem.

Some hybrid methods combining an exact method and a metaheuristic algorithm have been applied for solving CMND. Accordingly [Chouman and Crainic, 2010] generate a neighborhood structure based on a hybrid method combining an exact MIP method and a tabu search metaheuristic with an arc-balanced cycle procedure. [Yaghini et al. 2012a] proposed a new hybrid algorithm of simplex method and simulated annealing metaheuristic, the idea of which is to explore solution space more efficiently. Other hybrid approaches have been suggested for neighborhood structures [Yaghini and Kazemzadeh, 2012b; Yaghini et al., 2012b; Yaghini et al., 2013b; Yaghini et al., 2011].

This paper attempts to fill this gap by contributing towards reaching this goal. An MIP-ACO hybrid framework has been proposed as an approach to explore efficiently the solution space of CMND problems using the proposed ACO-based neighborhoods. The basic idea of the method is to construct a neighborhood solution based on ACO and using an MIP solver to determine the optimal amounts of flows to be transported on open arcs.

To verify the efficiency and effectiveness of the proposed algorithm, the obtained results are compared with existing approaches for a large set of benchmark instances. The statistical analysis shows that the proposed algorithm achieves the better result found in the literature.

This paper is organized as follows. In the next section, an MIP formulation of CMND is presented. Section 3 outlines the proposed ACO-based neighborhoods for CMND problem. Section 4 describes a solution method based on the proposed neighborhoods. The solution method parameters tuning, the obtained results, and statistical analysis are described in section 5. Finally, in section 6, conclusions are presented.

2. Mathematical Formulation

Given a directed graph G = (N, A), the set of nodes is N and the set of arc is A, which is assumed that all (i, j) A are design directed arcs. Let P denote the set of commodities to satisfy origin–destination pairs for each p P, and let represent the required amount of flow of commodity p to be transported from its (unique) origin o(p) to its (unique) destination s(p). The involved costs in the network are the transportation cost per unit of commodity p through the arc (i, j) which is denoted by , as well as the fixed cost of including arc (i, j) in the design of the network denoted by . On each arc (i, j) A, the total flow is limited by the capacity . To formulate the problem, two sets of decision variables are defined. (i, j) A signifies the design variables, and equals1, if arc (i, j) is selected in the final design and 0, otherwise. The continuous variables indicate the amount of flow of commodity p P on arc (i, j). The objective is to minimize the sum of transportation and fixed costs while transporting all the commodities. The model is formulated as follows.

$$\min \sum_{p\in P}\sum_{(i,j)\in A} c_{ij}^{p} x_{ij}^{p} + \sum_{(i,j)\in A} f_{ij}\, y_{ij} \tag{1}$$

$S.T$

$$\sum_{j\in N^{+}(i)} x_{ij}^{p} - \sum_{l\in N^{-}(i)} x_{li}^{p} = \begin{cases} d_p & if \quad i = O(p) \\ -d_p & if \quad i = S(p) \quad for\ all \quad p\in P \\ 0 & otherwise \end{cases} \tag{2}$$

$$\sum_{p\in P} x_{ij}^{p} \le u_{ij} y_{ij} \qquad for\ all \quad (i,j)\in A \tag{3}$$

$$x_{ij}^{p} \ge 0\ , \quad y_{ij} \in \{0,1\} \qquad for\ all\ (i,j)\in A,\ \ p\in P \tag{4}$$

Where the sets of outward and inward neighbors of any node are represented by and , respectively. The objective function (1) calculates the total system cost computed as sum of the fixed costs and the cost of shipping the demands on the final constructed network. Constraints (2) ensure that demands are satisfied between each origin–destination pair through equaling total incoming flow with total outgoing flow at every transshipment node. The linking constraints (3) state that the total flow (all commodities) on an open arc (y_{ij} =1) cannot exceed its capacity, while it must be 0 if the arc is closed (y_{ij} = 0). Relation (4) is the usual non-negativity and integrality constraints for decision variables.

3. The Proposed ACO-based Neighborhoods

The Ant Colony Systems (ACS), as a successful algorithm of ACO, is applied to construct the proposed neighborhood structure for solving CMND. Due to great speed of ACO in constructing feasible solutions, it is employed as an effective tool to search solution space. In the proposed neighborhoods, the joining solutions concept is inspired by RINS [Glover et al., 2009] and path relinking [Danna et al., 2005] algorithms.

In the following subsections, ACO, ACO-based neighborhoods, path construction method, a simple example, and a comparison of the proposed neighborhoods with RINS and path relinking are presented.

3.1 Ant Colony Optimization

[Dorigo et al. 1991] introduced the first ACO algorithm called Ant System (AS) in 1991 for solving Traveling Salesman Problem (TSP). The main source of inspiration of this optimization method is the behavior of ants searching food source from their nest through possible paths.

ACO algorithms are developed for many combinatorial problems such as vehicle routing problem, production scheduling, sequential ordering problem, and telecommunication routing to name but a few [Dorigo and Stutzle, 2004]. The ants are placed on a graph and forced to make a path from origin node to specific destination one. A feasible solution is generated by a set of ant's move.

Six years after instituting the first ACO, [Dorigo and Gambardella, 1997] introduced another algorithm that performed better than AS and called ACS. Different procedures in local and global updating of the pheromone trails are used in ACS. Many different versions of ACO algorithms such as Elitist Ant System, rank-based AS and Min–Max AS are described in [Stützle and Dorigo, 1999; Dorigo et al., 1999; Socha et al., 2003], sequentially.

3.2 ACO-based Neighborhoods

The method for constructing the proposed ACO-based neighborhood structure for CMND problem is illustrated by figure 1.

The proposed ACO-based neighborhood is commenced by transforming an open arc to

// ACO-BasedNeighborhoodMethod()

Input: incumbent solution;

Select and Close an open arc;

Set Pc = {the transported demands on the closed arc};

Set $_{n}$ = | P_c|;   // No. of demands in P_c

Update network arc capacities;

For (i = 1 **to** n)

    Set k = i^{th} demand from Pc;

    Call PathConstructionMethod(k);

End-For

Generate a sub-MIP model combining ant's and incumbent solutions;

Solve the generated sub-MIP model;

Close unused open arcs manually;

Output: the modified sub-MIP solution as a neighborhood solution;

Figure 1. Pseudo code for constructing the proposed neighborhood structure

close. In the previous works, many closing arcs strategies have been used [Ghamlouche et al., 2003; Ghamlouche et al., 2004, Crainic et al., 2006; Chouman and Crainic, 2010]. The method used in the proposed neighborhood, randomly chooses an open arc, and then closes it. In the next step, all shipped demands on the closed arc are identified and stored in set Pc, the stored commodities of which need to be removed from the other arcs of the network, eventually the arc capacities are required to be updated. At this step, there are possibly some open arcs with no commodity, so these arcs are identified and closed manually. For each commodity in set Pc, the ACO-based neighborhood structure constructs a feasible path by using PathConstructionMethod (PCM) to transport it. The path construction method will be explained in section 3.3.

In the next step, a new sub-MIP model is generated by combining the ACS and incumbent solutions, utilizing several new added constraints. Constraint (5) guarantees that all open arcs, with the exception of the one closed in the incumbent solution remain open, where A_O^{IC} is a subset of A that includes open arcs in the incumbent solution except the one closed. Constraint (6) enforces all of the closed arcs in both incumbent and ACS solutions, which remain closed, where A_C^{ICO} is a subset of A including the closed arcs in the ACS solution, and A_C^{IC} is a subset of A including the closed arcs in incumbent solution. Finally, constraint (7) ensures the value of the binary decision variable of the closed arc equals zero, in the generated sub-MIP model. The remaining arc variables are categorized as free variables and the MIP solver finds their best possible values. Figure 2 illustrates this process. Binary design variables determine state of each arc. Question marks in the generated sub-MIP solution vector are the values that an MIP solver is assumed to compute.

$$\sum_{(i,j)\in A_o^{IC}} y_{(i,j)} = | A_o^{IC} | \tag{5}$$

$$\sum_{(i,j)\in A_c^{IC} \cap A_c^{ICO}} y_{(i,j)} = 0 \tag{6}$$

$$y_{(i,j)} = 0 \qquad (i,j) \textit{ is the closed arc} \tag{7}$$

The ACS solution:	1	0	1	0	0	1	1
The incumbent solution:	1	1	0	1	0	1	0
				↓			
New Sub-MIP:	1	1	?	1	0	1	?

Figure 2. Binary design variable values in the combination process

In the next step, the created sub-MIP is solved by using an MIP solver and determined the optimal amounts of commodity flows on the arcs. Eventually, the open arcs which have no commodity on are identified and closed manually. The solution of this modified sub-MIP is considered as a neighborhood solution (neighborhoodsolution).

3.3 Path Construction

The method to construct a feasible path by a single ant is described in this section. The proposed ACO-based neighborhood structure is based on ACS algorithm in which an ant k is located at node i, and then an arc commencing with node i and terminating to node j is selected. Node j is chosen according to the pseudorandom proportional rule [Dorigo and Stutzle, 2004] given by following equations.

$$if \quad q \le q_0, \tag{8}$$

$$j = \begin{cases} \arg[Max_{l \in N_i^k} \{\tau_{il}[\eta_{il}]^\beta\}, \\ p_{ij}^k = \dfrac{[\tau_{il}]^\alpha[\eta_{ij}]^\beta}{\sum_{l \in N_i^k}[\tau_{il}]^\alpha[\eta_{ij}]^\beta}, \quad j \in N_i^k, \qquad else. \end{cases} \tag{9}$$

Where q is a random variable uniformly distributed in [0, 1], q0 (0 ≤ q0 ≤ 1) is a parameter, and p_{ij}^k is a random variable to decide which node to visit next. In other words, with probability q0, the ant makes the best possible move as indicated by the learned pheromone trails and the heuristic value, while with probability 1 – q0 it performs a biased exploration of the arcs. Tuning the parameter q0 allows modulation of the degree of exploration and the choice of whether to concentrate the search of the system around the best-so-far solution or to explore other solution space.

By the probabilistic rule (9), the probability of choosing a particular arc (i, j) increases with the value of the associated pheromone trail τ_{ij} and the heuristic information value η_{ij} [Dorigo and Stutzle, 2004].

The heuristic value η_{ij} = 1/c that is available a priori. If the arc (i, j) is opened on the incumbent solution, c equals to variable cost of the arc; otherwise, c is equal to sum of variable cost and per unit fixed cost. By this definition for η parameter, the algorithm intends to select an open arc to transport the commodity. The relative influence of the pheromone trail and the heuristic value is determined by the parameters α and β. N_i^k is the feasible neighborhood of ant k when at node i, that is to say, the set of nodes that ant k has not visited yet and the arcs ending to them have remaining capacity to satisfy demands (the probability of choosing a node outside N_i^k is 0).

After constructing a path for a commodity, arc capacities are updated as follows. Let remaining Volume be the remaining amount of commodity to be transported, and maxFlow given by $\min\{remainedVolume \cup \{\overline{u}_{ij}, \forall(i,j) \in newPath\}\}$ is the maximum flow that can be moved using the constructed new path, where is the remaining capacity of arc (i, j). The capacity of all arcs in the path needs to be updated by deducting them from maxFlow. In addition, the remaining volume of the commodity is updated by deducting maxFlow from remaining-Volume. By constructing new paths, the total volume of a commodity is transported from its origin to the destination. This procedure is iterated until all commodities in set Pc are delivered.

The pseudocode in figure 3 illustrates the procedure of path construction in the proposed

ACO-based neighborhood as shown in figure 1.

3.4 An Illustrative Example

For illustrating the proposed neighborhood structure, a simple example is presented in this section. The network in figure 4 is adopted from [Ghamlouche et al. 2003]. The labels of each arcs display their fixed cost, variable cost and capacity. The only two commodities are assumed in this network A→H with 2, and A→I with 3 volumes of demands, where all commodities are identified by origin-destination pairs.

An incumbent feasible solution, whose objective value is 73, is shown in figure 5. In figure

```
//  PathConstructionMethod(k)
//  new paths are generated for demand k
Input: k;
Set generatedPaths = {∅};
Set remainingCommodity = total  amount of commodity k;
While (remainingCommodity > 0)
    Set newPath = {origin(k)};
    Set currentNode = origin(k);
    While (currentNode # dest(k))
        Set randomValue = a random value between [0,1];
        If randomValue < q0 Then
            Select j based on arg[Max_{l∈N_i^k} {τ_il[η_il]^β}]
        Else
            Set p_ij^k = [τ_il]^α [η_ij]^β / Σ_{l∈N_i^k} [τ_il]^α [η_ij]^β
            Select j ∈ N_i^k based on p_ij^k ;
        End-If
        newPath = newPath + j;
        currentNode = j;
    End-While
    Set maxFlow = maximum flow that can be moved on newPath;
    generatedPaths = generatedPaths + newPath;
    Update network arc capacities;
    remainingVolume = remainingVolume – maxFlow;
End-While
Output: generatedPaths;
```

Figure 3. Pseudocode for constructing paths

5-a, the arc label shows the total flow passed. The amount of first commodity is transported through {(A-B), (B-F), (F-E), (E-H)} path, and the second commodity is passed via {(A-C), (C-F), (F-I)} and {(A-D), (D-G), (G-I)} paths. Figure 5-b illustrates the representation of the incumbent solution.

By assuming arc (B-F) is selected to close, arcs (A-B), (A-C), (A-D), (C-F), (D-G), (E-H), (F-I), (F-E) and (G-I) stay open. The first commodity is the only commodity on arc (B-F) which should be stored in set Pc. The amount of this commodity must be removed from the other arcs of the network and the network arc capacities need to be updated. By updating arc capacities, the arcs (A-B), (F-E) and (E-H) must be closed. By construction of new paths for A→H commodity, ACS algorithm creates a new solution (figure 6); additionally, the constructed network is given in figure 6-a.

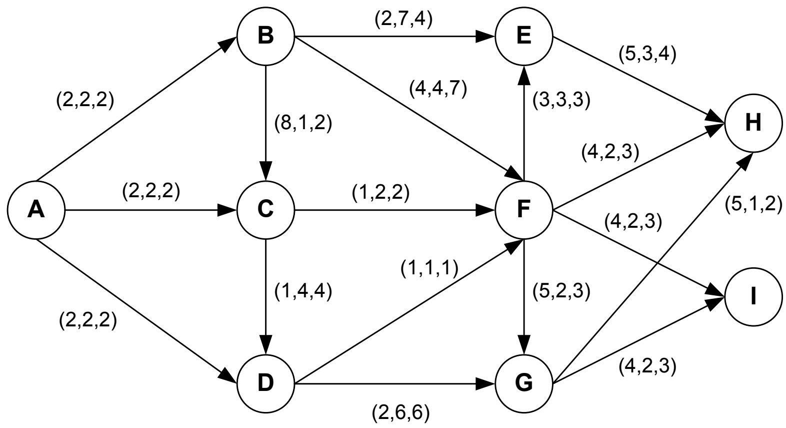
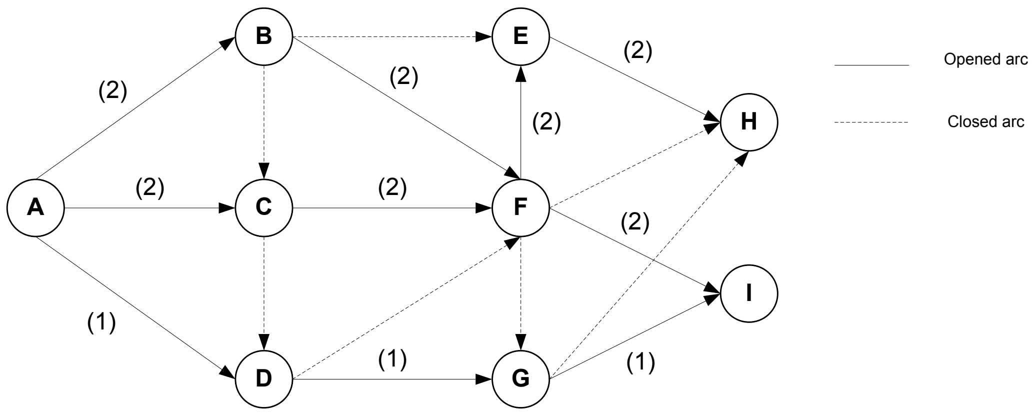


Figure 4. Physical network of the example



(5-a) The designed network of the incumbent solution

Arc	A-B	A-C	A-D	B-C	B-E	B-F	C-D	C-F	D-F	D-G	E-H	F-E	F-G	F-H	F-I	G-H	G-I
y_{ij}	1	1	1	0	0	1	0	1	0	1	1	1	0	0	1	0	1

(5-b) Solution representation

Figure 5. The incumbent solution

The ant first ships commodity through (A-D), (D-F) and (F-H) path, and then carries the remaining commodity on (A-B), (B-E) and (E-H) path. Figure 6-b shows the representation of the ACS solution.

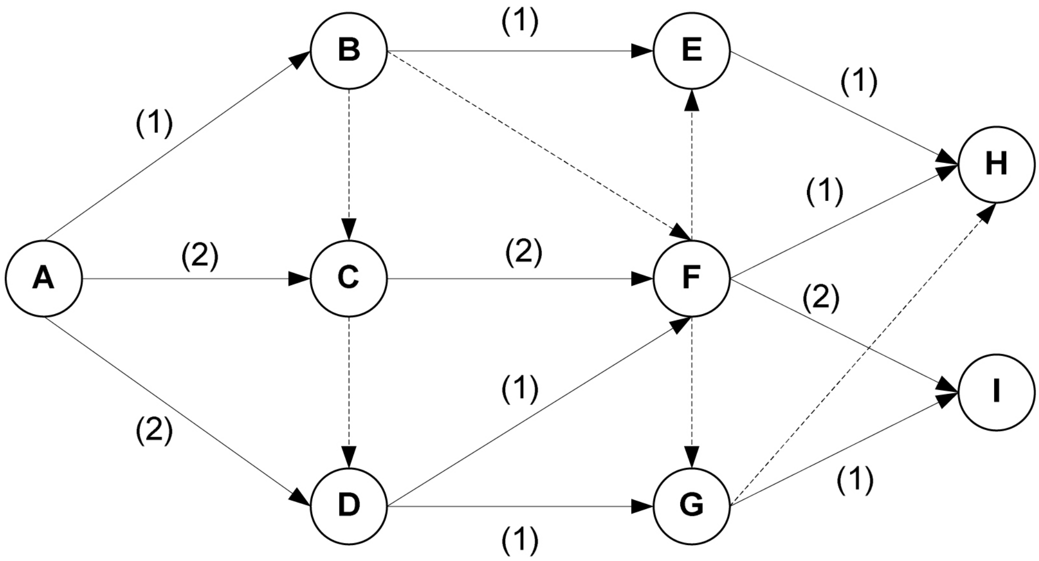
By joining the ACS and incumbent solutions, the open and closed arcs are determined on figure 7. Question marks in the generated sub-MIP model are the values that the MIP solver has to compute. Constraints (10), (11) and (12) are added to the sub-MIP model in line with constraints (5), (6) and (7).

$$y_{A-C} + y_{C-F} + y_{A-D} + y_{D-G} + y_{F-I} + y_{G-I} = 6 \quad (10)$$

$$y_{B-C} + y_{C-D} + y_{F-G} + y_{G-H} + y_{F-E} = 0 \quad (11)$$

$$y_{B-F} = 0 \quad (12)$$

After solving the sub-MIP model, the idle open arcs (G-I) and (D-G) must be closed.



(6-a) The constructed network of ACS solution

Arc	A-B	A-C	A-D	B-C	B-E	B-F	C-D	C-F	D-F	D-G	E-H	F-E	F-G	F-H	F-I	G-H	G-I
y_{ij}	1	1	1	0	1	0	0	1	1	1	1	0	0	1	1	0	1

(6-b) Solution representation

Figure 6. The ACS solution

Arc	A-B	A-C	A-D	B-C	B-E	B-F	C-D	C-F	D-F	D-G	E-H	F-E	F-G	F-H	F-I	G-H	G-I
y_{ij}	?	1	1	0	?	0	0	1	?	1	?	0	0	?	1	0	1

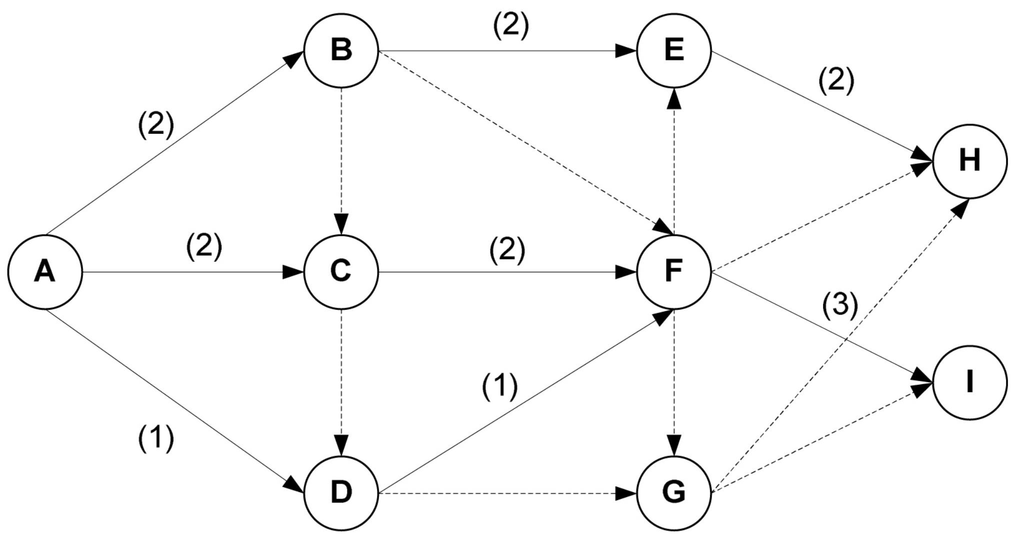
Figure 7. The binary variable values in the sub-MIP model

The first commodity is shipped through {(A-B), (B-E), (E-H)}, path and {(A-C), (C-F), (F-I)}, and {(A-D), (D-F), (F-I)} paths are used to transport the second commodity. The solution is considered a neighborhood solution whose objective value is 66 (figure 8).

3.5 ACO-based Neighborhoods vs. RINS and Path Relinking

The proposed method is related to path relinking [Glover et al. 2009] and Relaxation Induced Neighborhood Search (RINS) [Danna et al. 2005] approaches. It behaves like path relinking because, in a certain way, two elite solutions in the search space are relinked. Moreover, the proposed method is related to RINS because of combining the solution of a local search method and incumbent solutions to gain a neighborhood solution.

However, the proposed neighborhood differs from path relinking and RINS in the three following ways. The first and obvious difference is that RINS is based on a branch-and-cut tree, but the proposed neighborhood has been used in an MIP-metaheuristic framework. Second, the proposed ACO-based neighborhood structure searches the solution space by closing open arcs to reach the neighboring solution, whereas RINS performs the moves based on branch-and-cut procedure. Eventually, inasmuch as the neighborhood solution will be accepted only if its feasibility will be met



(8-a) The designed network of the sub-MIP model

Arc	A-B	A-C	A-D	B-C	B-E	B-F	C-D	C-F	D-F	D-G	E-H	F-E	F-G	F-H	F-I	G-H	G-I
y_{ij}	1	1	1	0	1	0	0	1	1	0	1	0	0	0	1	0	0

(8-b) Solution representation

Figure 8. The sub-MIP solution

through model constraints, it is impossible to find an infeasible neighborhood solution by path relinking algorithm while in the proposed neighborhood structure; hence, the generated neighborhood solution is always feasible.

4. The Proposed Algorithm using ACO-based Neighborhoods

In order to evaluate the proposed ACO-based neighborhood structure, a solution algorithm has been presented. Figure 9 illustrates the basics of the proposed solution algorithm starting with a feasible solution as an incumbent solution. The initial solution can be obtained by using a heuristic method or an MIP solver in a limited time. After setting the initial parameters, the main loop undergoes repetition until meeting termination condition. In the inner loop which is started by calling ACO_BasedNeighborhoodMethod (BNM), neighborhoodsolution will be constructed and its objective value is compared with the incumbent objective value. If the neighborhood solution improves the incumbent objective value, the neighborhoodsolution is accepted as a new incumbent solution, all of the neighborhood structure parameters are reinitialized, and the loop is repeated. In addition, if the neighborhood objective value is bigger than incumbent solution objective value, the inner loop iterates maximum number of ants in a colony (antMax).

After building neighborhood solution as well as comparing it with incumbent solution, the pheromone trail on each arc for each commodity is updated. The data structure of the pheromone trail archive is shown in figure 10.

Where indicates the learned pheromone trail for commodity i on arc j. At the initiation time of the algorithm, the pheromone trail on each arc for each commodity is initialized. In the proposed algorithm, the pheromone trails are allowed to be added to the best-so-far solution at the end of each iteration. The pheromone trails are updated using equations (13) and (14).

$$\tau_i^j \rightarrow \begin{cases} (1-\rho)\tau_i^j + \rho x_i^j \Delta\tau^{bs}, & \forall x_i^j > 0, \quad \text{(13) Eq.} \\ (1-\rho)\tau_i^j, & else. \quad \text{(14) Eq.} \end{cases}$$

Where $\Delta\tau^{bs} = \gamma / C^{bc}$, C^{bc} is the objective value of the best-so-far solution, and γ is an input parameter. x_i^j is decision variable that shows the amount of flow commodity i on arc j. In the proposed algorithm, evaporation applies to all of the arcs and the new pheromone is deposited at the arcs in the best-so-far solution. The deposited pheromone is discounted by the decision variable x_i^j and the factor $(0 < \rho < 1)$.

The proposed algorithm has many practical applications in a wide variety of large networks such as road network, air transportation and especially railways transportation networks; because the realistic practical problems have extensive amounts of variables and

```
// Proposed Algorithm with ACO-Based Neighborhood Structure
Input: INITIAL_Q0 and ANT_MAX and MAX_Q0_ DECREASE;
Generate initialSolution;
Set  incumbentSolution = initialSolution;
Set  incumbentObj = initialObj;
Set  currentQ0 = INITIAL_Q0;
Set  antCounter = 0;
Set  decreaseQ0Counter = 0;
Repeat
     While (antCounter is smaller than antMax)
          Call ACO-BasedNeighborhoodMethod();
          If (neighborhoodsolution is better than incumbentSolution)
               Set currentQ0 = INITIAL_Q0;
               Set antCounter = 0;
               Set decreaseQ0Counter = 0;
               Set incumbentSolution = neighborhoodsolution;
               Set incumbentObj = neighborhoodObj;
          Else
               Set  antCounter = antCounter +1;
          End-if
          Update pheromone trail;
     End-whie
Until (currentQ0 < (INITIAL_Q0 - (decreaseQ0Counter * MAX_Q0_ DECREASE)))
Output: incumbentSolution and incumbentObj;
```

Figure 9. Pseudocode for the proposed algorithm using ACO-based neighborhoods

Arc / *Commodity*	*1*	*2*	...	*j*	...	*n*
1	τ_1^1	τ_1^2	...	τ_1^j	...	τ_1^n
2	τ_2^1	τ_2^2	...	τ_2^j	...	τ_2^n
⋮	⋮	⋮	⋱	⋮	⋱	⋮
i	τ_i^1	τ_i^2	...	τ_i^j	...	τ_j^n
⋮	⋮	⋮	⋱	⋮	⋱	⋮
k	τ_k^1	τ_k^2	...	τ_k^j	...	τ_k^n

Figure 10. The pheromone trail data structure

constraints, which require lots of times and costs for optimally solving.

5. Computational Results

5.1 Parameter Tuning

Metaheuristic developers adjust the parameters one by one, so the best values are determined experimentally. In this approach, not only the interaction between parameters is ignored, but finding optimal values is unable to guarantee as well [Talbi, 2009]. In this paper, in order to accomplish this problem, statistical Design of Experiments (DOE) [Montgomery, 2009] and Design Expert software are used for tuning the proposed algorithm parameters. There are many works using DOE method such as [Coy et al., 2001; Ridge and Kudenko, 2010] for parameter tuning. In this paper, a three-step method on the basis of previous works on parameter tuning using DOE is applied. The stages of the approach are (1) parameter screening, (2) response surface modeling, and (3) optimization of parameters. Step 1 screens the effective and important parameters on algorithm responses. In the next step, a mathematical function relating the tuning parameters to each of the response of interest is given. And ultimately, the parameters are optimized with respect to importance of each response.

In the proposed algorithm, solution quality and CPU time are considered as the response variables. The tuned parameters levels and values for solving problems are illustrated in table 1.

5.2 Results

In analyzing the results, the outcome of the proposed algorithm using the ACO-based neighborhoods is compared with the results of best five methods in the literature [Ghamlouche et al., 2003; Ghamlouche et al., 2004; Crainic et al., 2006; Martín and González, 2010; Chouman and Crainic, 2010]. To ensure meaningful comparisons, performance of the algorithm was tested on a set of 43 benchmark instances described in [Gendron and Crainic, 1994; Gendron and Crainic, 1996]. The best

Table 1. Level factorial design for the proposed algorithm

Factor	Level		Final parameter
	Low	High	
α	1.00	5.00	5.00
β	0.25	2.00	1.00
ρ	0.00	0.10	0.01
INITIAL_Q0	0.80	1.00	0.95
ANT_MAX	5.00	40.00	10.00
MAX_Q0_ DECREASE	5.00	10.00	10.00

five methods have used the same instances.

The benchmark instances have some characteristics. Each arc is associated with positive variable as well as fixed costs, and transportation capacity. The origin and destination of each demand is presented. The first column in table 2 describes instances by the number of nodes (|N|), the number of arcs (|A|), and the number of commodities (|P|), and the fixed cost and capacity information. The comparison between the fixed and variable costs to show their relative dominance is identified by F and V, respectively. Compared to the total demand, T identifies that the problem is tightly capacitated, and L shows it is loose. There are detailed descriptions of the instances in [Gendron and Crainic, 1994; Gendron and Crainic, 1996]. The proposed algorithm has been coded in Java programming language. The compared results are summarized in table 2, the first column of which shows the problem name and its quintuplet features. The Obj Val columns show the solution values, and Time column displays running times in seconds. In table 2, TC stands for the cycle-based tabu search [Ghamlouche et al., 2003], PR for the path relinking procedure [Ghamlouche et al., 2004], MLEVEL for the multilevel cooperative heuristic [Crainic et al., 2006], LOCB for the local branching algorithm [Martín and González, 2010], MT for the MIP-Tabu search [Chouman and Crainic, 2010], and finally, ACO-NS stands for the proposed ACO-based neighborhood structure.

Table 3 illustrates the samples in which the proposed algorithm can obtain better solutions the other ones. These improvements can be seen as negative gap values in the table. The percentage of improvement of the proposed algorithm relative to the best solution given by any of the other five methods is displayed in table 3. Negative entries in column indicate better performance in solution quality found by the proposed algorithm.

The results shown in table 3 prove that the proposed neighborhood structure can both efficiently and accurately search the solution space of CMND. Table 4 indicates the proposed algorithm improves the best solution values for 19 problems solved by the previous methods, and for 13 problems, the proposed algorithm finds the best solution found by the others.

5.3 Statistical Analysis

To evaluate the computational results, SPSS Statistics 16 software is used to perform a statistical analysis. For parametric statistical tests, a data normality assumption is required. After normalizing datasets, the test of normality called Shapiro-Wilk test is used. Next, an analysis of variance (ANOVA) is performed [Montgomery and Runger, 2006; Freund, 1992].

The significance level is considered 5 percent (α=0.05). The following hypothesis is tested:

H_0: The difference between the proposed al-

Table 2. the compared results of the proposed algorithm

Instance	TC	PR	MLEVEL	LOCB	MT	ACO-NS	
	Obj Val	Obj Val	Obj Val	Obj Val	Obj Val	Time(s)	Obj Val
20,230,40,V,L	430628.0	424385.0	426702.0	423848.0	423848.0	6.3	423848.0
20,230,40,V,T	372522.0	371811.0	371475.0	371475.0	371475.0	8.9	371475.0
20,230,40,F,T	652775.0	645548.0	652894.0	643036.0	643538.0	23.3	643036.0
20,230,200,V,L	100001.0	100404.0	98582.0	95295.0	94218.0	2306.0	94213.0
20,230,200,F,L	148066.0	147988.0	143150.0	143446.0	138491.0	2668.0	138109.0
20,230,200,V,T	106868.0	104689.0	102030.0	98039.0	98612.0	1413.8	97914.0
20,230,200,F,T	147212.0	147554.0	141188.0	141128.0	136309.0	5216.0	137162.5
20,300,40,V,L	432007.0	429398.0	429837.0	429398.0	429398.0	6.6	429398.0
20,300,40,F,L	602180.0	590427.0	593544.0	586077.0	588464.0	16.2	586077.0
20,300,40,V,T	466115.0	464509.0	466004.0	464509.0	464509.0	10.8	464509.0
20,300,40,F,T	615426.0	609990.0	619203.0	604198.0	604198.0	11.8	604198.0
20,300,200,V,L	81367.0	78184.0	78209.5	76375.0	75045.0	2646.0	75084.0
20,300,200,F,L	122262.0	123484.0	121951.0	119142.8	116259.0	2564.0	116861.0
20,300,200,V,T	80344.0	78866.8	77251.0	76167.5	74995.0	2750.0	74991.0
20,300,200,F,T	113947.0	113584.0	111173.0	109808.0	109164.0	1988.2	107169.0
30,520,100,V,L	56603.0	54904.0	55754.0	54026.0	54008.0	672.9	53966.0
30,520,100,F,L	103657.0	102054.0	99817.0	96255.0	93967.0	1983.0	94653.0
30,520,100,V,T	54454.0	53017.0	53512.0	52129.0	52156.0	751.1	52079.0
30,520,100,F,T	105130.0	106130.0	102477.0	101102.0	97490.0	964.0	97581.3
30,520,400,V,L	122673.0	119416.0	115671.0	114367.4	112927.0	3303.0	113018.6
30,520,400,F,L	164140.0	163112.0	156601.0	157725.5	149920.0	3723.0	149944.0
30,520,400,V,T	122655.0	120170.0	120980.0	115240.0	114664.0	7801.7	115038.0
30,520,400,F,T	169508.0	163675.0	160217.0	168561.0	152929.0	13257.6	159919.5
30,700,100,V,L	50041.0	48723.0	48869.0	47603.0	47603.0	114.3	47603.0
30,700,100,F,L	64581.0	63091.0	63756.0	60272.0	60184.0	700.4	60184.0
30,700,100,V,T	48176.0	47209.0	47457.0	45905.0	45880.0	1922.0	45875.0
30,700,100,F,T	57628.0	56575.5	56910.0	55104.0	54926.0	801.1	54904.0
30,700,400,V,L	107727.0	105116.0	102631.0	103787.0	97982.0	3477.0	97808.6
30,700,400,F,L	150256.0	145026.0	143988.0	169759.7	135109.0	12961.0	137539.7
30,700,400,V,T	101749.0	101212.0	99194.9	96680.0	95781.0	5311.2	95271.0
30,700,400,F,T	144852.0	141013.0	138266.0	144925.5	130856.0	10800.0	130106.9
25,100,10,V,L	14712.0	14712.0	14712.0	14712.0	-	5.7	14712.0
25,100,10,F,L	14941.0	14941.0	14941.0	14941.0	-	27.5	14941.0
25,100,10,F,T	50529.0	49899.0	49937.0	49899.0	-	67.5	49899.0
25,100,30,V,T	365385.0	365385.0	365385.0	365272.0	-	56.8	365272.0
25,100,30,F,L	37515.0	37654.0	37607.0	37325.5	-	10800.0	37055.0
25,100,30,F,T	87325.0	86428.0	86461.3	85530.0	-	101.4	85530.0
100,400,10,V,L	28786.0	28485.0	28553.0	28423.0	28423.0	7.1	28423.0
100,400,10,F,L	24022.0	24022.0	24022.0	24690.0	24161.0	2833.7	23949.0
100,400,10,F,T	67184.0	65278.0	66284.0	67357.0	67233.0	10800.0	65187.0
100,400,30,V,T	385508.0	384926.0	385282.0	384809.0	384940.0	691.4	384802.0
100,400,30,F,L	51831.0	51325.0	50456.0	49872.0	49682.0	699.3	50012.0
100,400,30,F,T	147193.0	141359.0	145721.0	141633.0	144349.0	10800.0	139380.0

Table 3. the relative gaps between the proposed method and the other methods

Instance	TC	PR	MLEVEL	LOCB	MT
20,230,40,V,L	-1.57%	-0.13%	-0.67%	0.00%	0.00%
20,230,40,V,T	-0.28%	-0.09%	0.00%	0.00%	0.00%
20,230,40,F,T	-1.49%	-0.39%	-1.51%	0.00%	-0.08%
20,230,200,V,L	-5.79%	-6.17%	-4.43%	-1.14%	-0.01%
20,230,200,F,L	-6.72%	-6.68%	-3.52%	-3.72%	-0.28%
20,230,200,V,T	-8.38%	-6.47%	-4.03%	-0.13%	-0.71%
20,230,200,F,T	-6.83%	-7.04%	-2.85%	-2.81%	0.63%
20,300,40,V,L	-0.60%	0.00%	-0.10%	0.00%	0.00%
20,300,40,F,L	-2.67%	-0.74%	-1.26%	0.00%	-0.41%
20,300,40,V,T	-0.34%	0.00%	-0.32%	0.00%	0.00%
20,300,40,F,T	-1.82%	-0.95%	-2.42%	0.00%	0.00%
20,300,200,V,L	-7.72%	-3.97%	-4.00%	-1.69%	0.05%
20,300,200,F,L	-4.42%	-5.36%	-4.17%	-1.92%	0.52%
20,300,200,V,T	-6.66%	-4.91%	-2.93%	-1.54%	-0.01%
20,300,200,F,T	-5.95%	-5.65%	-3.60%	-2.40%	-1.83%
30,520,100,V,L	-4.66%	-1.71%	-3.21%	-0.11%	-0.08%
30,520,100,F,L	-8.69%	-7.25%	-5.17%	-1.66%	0.73%
30,520,100,V,T	-4.36%	-1.77%	-2.68%	-0.10%	-0.15%
30,520,100,F,T	-7.18%	-8.05%	-4.78%	-3.48%	0.09%
30,520,400,V,L	-7.87%	-5.36%	-2.29%	-1.18%	0.08%
30,520,400,F,L	-8.65%	-8.07%	-4.25%	-4.93%	0.02%
30,520,400,V,T	-6.21%	-4.27%	-4.91%	-0.18%	0.33%
30,520,400,F,T	-5.66%	-2.29%	-0.19%	-5.13%	4.57%
30,700,100,V,L	-4.87%	-2.30%	-2.59%	0.00%	0.00%
30,700,100,F,L	-6.81%	-4.61%	-5.60%	-0.15%	0.00%
30,700,100,V,T	-4.78%	-2.83%	-3.33%	-0.07%	-0.01%
30,700,100,F,T	-4.73%	-2.95%	-3.52%	-0.36%	-0.04%
30,700,400,V,L	-9.21%	-6.95%	-4.70%	-5.76%	-0.18%
30,700,400,F,L	-8.46%	-5.16%	-4.48%	-18.98%	1.80%
30,700,400,V,T	-6.37%	-5.87%	-3.96%	-1.46%	-0.53%
30,700,400,F,T	-10.18%	-7.73%	-5.90%	-10.22%	-0.57%
25,100,10,V,L	0.00%	0.00%	0.00%	0.00%	-
25,100,10,F,L	0.00%	0.00%	0.00%	0.00%	-
25,100,10,F,T	-1.25%	0.00%	-0.08%	0.00%	-
25,100,30,V,T	-0.03%	-0.03%	-0.03%	0.00%	-
25,100,30,F,L	-1.23%	-1.59%	-1.47%	-0.72%	-
25,100,30,F,T	-2.06%	-1.04%	-1.08%	0.00%	-
100,400,10,V,L	-1.26%	-0.22%	-0.46%	0.00%	0.00%
100,400,10,F,L	-0.30%	-0.30%	-0.30%	-3.00%	-0.88%
100,400,10,F,T	-2.97%	-0.14%	-1.65%	-3.22%	-3.04%
100,400,30,V,T	-0.18%	-0.03%	-0.12%	0.00%	-0.04%
100,400,30,F,L	-3.51%	-2.56%	-0.88%	0.28%	0.66%
100,400,30,F,T	-5.31%	-1.40%	-4.35%	-1.59%	-3.44%

Table 4. Comparison of relative GAPs between the proposed algorithm and other methods

	TC	PR	MLEVEL	LOCB	MT
Average GAP(%)	-4.37%	-3.09%	-2.51%	-1.80%	-0.08%
Number of Improve	41	38	40	28	18

gorithm and local branching method (LB) is not significant (*μProposed algorithm = μLB*),
H_1: The difference between the proposed algorithm and local branching method (LB) is significant (*μProposed algorithm ≠ μLB*).

Table 5 shows the output table of SPSS Statistics software for ANOVA test. The obtained p-value for the test is 0.001, and it is smaller than significance level (0.001 < 0.05). At α=0.05, there is enough evidence to conclude that there is a significant difference in the relative gap of the proposed and local branching algorithms, and the proposed algorithm improves the solutions compared to local branching method. The ANOVA test, therefore, shows that the proposed algorithm is more effective than local branching method to solve CMND problem.

Moreover, the following hypothesis is tested:
H0: The difference between the proposed algorithm and MIP-Tabu search method is not significant (*μProposed algorithm = μMIP-Tabu*),
H1: The difference between the proposed algorithm and MIP-Tabu search method is significant (*μProposed algorithm ≠ μMIP-Tabu*).
The obtained p-value (0.703) is not smaller than significance level (0.05) so the difference between the proposed algorithm and MIP-Tabu search method is not significant (H0 is accepted). However, by considering the obtained gap between the proposed algorithm and MIP-Tabu search method, the proposed algorithm improves the results of this method

Table 5. Output table of ANOVA test for comparing the results of the proposed algorithm to other methods

Source of variation	Sum of Squares	Df	Mean Square	F	Sig.
Between Groups	69.606	1	69.606	11.979	.001
Within Groups	488.096	84	5.811		
Total	557.702	85			

in general.

6. Conclusions

In this paper, ACO-based neighborhoods to solve CMND problem are presented. The proposed neighborhoods are developed based on joining solutions of incumbent and ACO solutions. In order to assess the proposed neighborhoods, a solution algorithm is developed. The validity and efficiency of the proposed algorithm are tested on several standard test problems. To adjust the algorithm parameters, DOE method is used. The experimental results and statistical analysis show the efficiency and effectiveness of the proposed algorithm. The proposed algorithm improves the objective value of most instances in relation to local branching and MIP-Tabu search algorithms as the best methods in the literature. The proposed algorithm improves their results by -1.80 and -0.08, respectively. The outcome is that the proposed algorithm outperforms other methods in the literature, providing the best solution for the most benchmark instances with a reasonable computational effort.

All in all, one of the major future research, which is presumably considered, will be to apply other metaheuristics for generating paths, namely, Tabu Search, Simulated Annealing and Genetic Algorithm to name but a few.

Table 6. Output table of ANOVA test

Source of variation	Sum of Squares	Df	Mean Square	F	Sig.
Between Groups	.107	1	.107	.147	.703
Within Groups	52.300	72	.726		
Total	52.406	73			

7. References

-Balakrishnan, A., Magnanti, T. L. and Mirchandani, P. (1997) "Chapter: Network design", In: (1997) Annotated Bibliographies in Combinatorial Optimization". Ed. Dell'Amico, M.
Maffioli, F. and Martello, S., New York : John Wiley and Sons, pp. 311-329.

- Chouman, M. and Crainic, T. G. (2010) "An MIP-Tabu search hybrid framework for multicommodity capacitated fixed-charge network design", In: CRT-2010, Ed. 31th, CIRRELT, Université de Montréal.

-Chouman, M., Crainic, T. G. and Gendron,

B. (2003) "A cutting-plane algorithm based on cut set inequalities for multi commodity capacitated fixed charge network design", In: CRT-2003, Ed. 16th, Centre der Echerchesur-les Transports, Universitéde Montréal.

-Chouman, M., Crainic, T. G. and Gendron, B. (2009) "A cutting-plane algorithm for multi-commodity capacitated fixed-charge network design", In: CIRRELT, Centre de Recherche sur les Transports, Universite de Montreal, Montreal.

- Costa, A. M. (2005) "A survey on benders decomposition applied to fixed-charge network design problems". Computers & Operations Research, Vol. 32, No. 6, pp. 1429–1450.

- Coy, S., Golden, B., Runger, G. and Wasil, E. (2001) "Using experimental design to find effectiveparameter settings for heuristics", Journal of Heuristics, Vol. 7, No. 1, pp. 77-97.

- Crainic, T.G. (1999) "Network design in freight transportation". European Journal of Operational Research, Vol. 122, No. 2, pp. 272-288.

- Crainic, T.G. and Gendron, B. (2002) "Co-operative parallel tabu search for capacitated network design". Journal of Heuristics, Vol. 8, No. 6, pp. 601-627.

- Crainic, T.G., Frangioni, A. and Gendron, B. (2001) "Bundle-based relaxation methods for multicommodity capacitated fixed charge network design". Discrete Applied Mathematics, Vol. 112, No. 1, pp. 73-99.

- Crainic, T.G., Gendreau, M. and Farvolden, J.M. (2000) "A simplex-based tabu search method for capacitated network design". INFORMS Journal on Computing, Vol. 12, No.3, pp. 223–236.

- Crainic, T.G., Gendreau, M. and Hernu, G. (2004) "A solpe scaling/ Lagrangian perturbation heuristic with long-term memory for multi commodity capacitated fixed-charge network design". Journal of Heuristics, Vol.10, No. 5, pp. 525-45.

- Crainic, T.G., Li, Y. and Toulouse, M. (2006) "A first multi level cooperative algorithm for capacitated multi commodity network design". Computers & Operations Research, Vol. 33, No. 9, pp. 2602–2622.

- Danna, E., Rothberg, E. and Le Pape, C. (2005) "Exploring relaxation induced neighborhood s to improve MIP solutions". Mathematical Programming, Vol. 102, No. 1, pp. 71-90.

- Dorigo, M. and Gambardella, L.M. (1997) "A cooperative learning approach to the traveling salesman problem". IEEE Transaction on Evolutionary Computation, Vol. 1, No. 1, pp. 53–66.

- Dorigo, M., Maniezzo, V. and Colorni, A. (1991) “Ant system: An autocatalytic optimizing process”, Dipartimento di Elettronica e Informazione, Politecnico di Milano.

- Dorigo, M. and Stutzle, T. (2004) “Ant colony optimization”, MIT Press Cambridge.

- Dorigo, M., Caro, G. D. and Gambardella, L. M. (1999) “Ant algorithms for discrete optimization”. Artificial Life, Vol. 5, No. 2, pp. 137–172.

- Freund, J. E. (1992) “Mathematical statistics”. 5th ed. Prentice-Hall, New Jersey.

- Gendron, B. and Crainic, T.G. (1994) “Relaxations for multi commodity capacitated network design problems”. In: CRT-94, Ed. 5th, Centre der Echerchesurles Transports, Université de Montréal.

- Gendron, B. and Crainic, T. G. (1996) “Bounding procedures for multicommodity capacitated fixed charge network design problems”. In: CRT-96, Ed. 6th. Centre de Recherche sur les Transports, Universite de Montreal.

- Ghamlouche, I., Crainic, T. G. and Gendreau, M. (2003) “Cycle-based neighborhoods for fixed-charge capacitated multi commodity network design”. Operations Research, Vol. 51, No. 4, pp. 655–667.

- Ghamlouche, I., Crainic, T.G. and Gendreau, M. (2004) “Path relinking, cycle-based neighborhoods and capacitated multi commodity network design”. Annals of Operations Research, Vol. 131, No. 1-4, pp. 109–133.

- Glover, F., Laguna, M. and Martí, F. (2009) “Fundamentals of scatter search and path relinking”. Control and Cybernetics, Vol. 29, No. 3, pp. 653-684.

- Holmberg, K. and Yuan, D. (2000) “A Lagrangian heuristic based branch-and-bound approach for the capacitated network design problem”. Operations Research, Vol. 48, No. 3, pp. 461-481.

- Magnanti, T.L. and Wong, R.T. (1986) “Network design and transportation planning: Models and algorithms”. Transportation Science, Vol. 18, No.1, pp. 1-55.

- Martín, I.R. and González, J.J.S. (2010) “A local branching heuristic for the capacitated fixed-charge network design problem”. Computers & Operations Research, Vol. 37, No. 3, pp. 575-581.

Minoux, M. (1986) “Network synthesis and optimum network design problems: Models, solution methods and applications”. Networks, Vol. 19, No. 3, pp. 313-360.

- Montgomery, D.C. (2009) "Design and Analysis of Experiments", 7th ed., John Wiley and Sons.

- Montgomery, D. C. and Runger, G. (2006) "Applied statistics and probability for engineering". 3rd ed., New York: John Wiley.

- Ridge, E. and Kudenko, D. (2010) "Tuning an algorithm using design of experiments". In: Experimental methods for the analysis of optimization algorithms, Ed. Bartz-Beielstein, T., Chiarandini, M., Paquete, L. and Preuss, M. Springer Berlin Heidelberg. pp. 265–286.

- Socha, K., Sampels, M. and Manfrin, M. (2003) "Ant algorithms for the university course timetabling problem with regard to the state-of-the-art". In: Proceedings of the 3rd European Workshop on Evolutionary Computation in Combinatorial Optimization (EvoCOP), UK. pp. 334–345.

- Stützle, T. and Dorigo, M. (1999) "ACO algorithms for the traveling salesman problem". Evolutionary Algorithms in Engineering and Computer Science, pp. 163-183.

- Talbi, E. G. (2009) "Metaheuristics: From design to implementation", New York: John Wiley & Sons.

- Yaghini, M. and Kazemzadeh, M. R. A. (2012a) "Multicommodity network design problem in rail freight transportation planning". In: Paper presented at the 8th International Conference on Traffic and Transportation Studies, Changsha, China.

- Yaghini, M. and Kazemzadeh, M. R. A. (2012b) "A simulated annealing algorithm for unsplittable capacitated network design". International Journal of Industrial Engineering, Vol. 23, No. 2, pp. 91-100.

- Yaghini, M., Ahmadi, H. R., Barati, E., and Saghian, Z. (2013) "A tabu search algorithm for the railroad blocking problem". Journal of Transportation Engineering, Vol. 139, No. 2.

- Yaghini, M., Karimi, M., Rahbar, M. and Akhavan, R. (2011) "A hybrid simulated annealing and simplex method for fixed-cost capacitated multicommodity network design". International Journal of Applied Metaheuristic Computing , Vol. 2, No. 4, pp. 13-28.

- Yaghini, M., Momeni, M. and Sarmadi, M. (2012a) "A simplex–based simulated annealing algorithm for node–arc capacitated multicommodity network design". Applied Soft Computing, Vol. 12, No. 9, pp. 2997-3003.

- Yaghini, M., Momeni, M. and Sarmadi, M. (2012b). "Solving train formation problem using simulated annealing algorithm in a simplex framework". Journal of Advanced Transportation (in press).

- Yaghini, M., Momeni, M. and Sarmadi, M. (2013a). “An improved local branching approach for train formation planning”. Applied Mathematical Modelling, Vol. 37, No. 4, pp. 2300–2307.

- Yaghini, M., Rahbar, M. and Karimi, M. (2013b). “A hybrid simulated annealing and column generation approach for capacitated multicommodity network design”. Journal of the Operational Research Society, Vol. 64, No. 7, pp. 1010–1020.