A Simulated Annealing Algorithm for Unsplittable Capacitated Network Design

M. Yaghini[*] & M.R. Akhavan Kazemzadeh

Masoud Yaghini, School of Railway Engineering, Iran University of Science and Technology

Mohammad Rahim Akhavan, Department of Railway Transportation Engineering, Kermanshah University of Science and Technology

KEYWORDS

ABSTRACT

The Network Design Problem (NDP) is one of the important problems in combinatorial optimization. Among the network design problems, the Multicommodity Capacitated Network Design (MCND) problem has numerous applications in transportation, logistics, telecommunication, and production systems. The MCND problems with splittable flow variables are NP-hard, which means they require exponential time to be solved in optimality. With binary flow variables or unsplittable MCND, the complexity of the problem is increased significantly. With growing complexity and scale of real world capacitated network design applications, metaheuristics must be developed to solve these problems. This paper presents a simulated annealing approach with innovative representation and neighborhood structure for unsplittable MCND problem. The parameters of the proposed algorithms are tuned using Design of Experiments (DOE) method and the Design-Expert statistical software. The performance of the proposed algorithm is evaluated by solving instances with different dimensions from OR-Library. The results of the proposed algorithm are compared with the solutions of CPLEX solver. The results show that the proposed SA can find near optimal solution in much less time than exact algorithm.

1. Introduction

The network design problem is one of the important problems in combinatorial optimization. The objective of NDP is to find a minimum cost network on the available arcs for each commodity, which satisfy flows of commodities. Network design models have numerous applications in various fields such as transportation [38] [6], telecommunication [40] [20]; [41], and distribution planning [33].

---
[*] Corresponding author: Masoud Yaghini

Email: yaghini@iust.ac.ir
Paper first received Mar. 10, 2011, and in revised form Dec. 24, 2011.

One type of the network design models is splittable Multicommodity Capacitated Network Design (MCND) Problem. In this model, multiple commodities such as goods, data, people, etc., must be routed between different points of origin and destination on the arcs with limited capacity. In this problem, in addition to a unit cost (variable cost), a fixed cost is usually added due to the opening cost for the first time the arc is used. The MCND problem seeks a network with minimum cost which satisfies demands of commodities. This minimum cost is the sum of the fixed and variable cost [26]. In contrast with splittable MCND, flows of commodities cannot be separated on different routes in unsplittable MCND [38].

Network design problems can be easy stated but solving them is too difficult [2]. There are effective exact solution methods for uncapacitated network design problem. In exact algorithms, finding optimal solution is guaranteed.

Benders decomposition and branch-and-bound methods have been the two most effective methods for this type of problem [38]. The linear programming relaxations do not provide good lower bounds, and therefore simplex-based branch-and-bound methods are unlikely to succeed in solving even small instances. Indeed, most of the literature on exact approaches for the MCND problems and related problems uses Lagrangian relaxation [8]. Several surveys on network design models and their exact solution methods can be found in [38], [40], [2], [8], [42], and [17]. In addition, in [20], a good survey on models and solution methods for multicommodity capacitated network design problems, especially simplex-based cutting plane and Lagrangian relaxation solution methods has been presented.

Adding capacity to the arcs of network design problem adds more complexity to this problem [2]. Furthermore, large-scale network design problems, which occur in real world applications, are very difficult to solve [38]. There are theoretical and empirical evidence that the capacitated network design problems are NP-hard [38] [2]. It means that there is no efficient algorithm, which can solve them in polynomial time.

As a result, the researchers proposed approximate methods to solve them. However, this type of solution methods cannot guarantee the optimality of solutions. Heuristics and metaheuristics are the two classes of approximate methods [50]. Recent research works on developing approximate methods for multicommodity network design problem, can be found in [9], [10], [35], and [39].

Heuristic solution method more often stop at the local optima. Hence, metaheuristics deal with these problems by introducing systematic rules to escape from local optima [14] [12] [44] [29].

The popular metaheuristics for MCND problems include Genetic Algorithm (GA) [32] [30] [15], Ant Colony Optimization (ACO) [13] [14], Tabu Search (TS) [27] [28], and Simulated Annealing (SA) [36]. Over the past two decades, Genetic algorithms have been become as one of the solution method for some of variants of network design problem [21], [22] [23] [24] [25] [37].

Since ACO concepts are suitable for network problems, one can find applications of that for MCND problems [48] [45]. In the case of TS algorithm, there are also several research works on MCND problems including [7], [26] [53], [46], [31], and [5].

The SA algorithm is another metaheuristic that is used for solving the MCND. In [33] a SA solution method for distribution network problem was proposed. In [48] a hybrid SA and ACO approach for solving transportation network design was proposed. In [47] a simulated annealing method for reverse logistic network design problem was proposed. In [34] a SA method for logistic network design problem was proposed. Other applications of SA in computer network design can be found in [52] and [16].

This paper proposes a SA algorithm as a metaheuristic solution method for unsplittable MCND problem. In addition, an innovative representation is proposed for representing solutions. Several instances are selected from OR-Library [3] for performance evaluation of proposed algorithms. The parameter tuning for the proposed algorithm is done by means of Design of Experiments (DOE) approach and Design-Expert statistical software [51]. The results of the algorithm are compared with CPLEX solver with branch-and-bound method.

The paper is organized as follows. In the next section, unsplittable MCNDP optimization model is presented. In section 3, the SA solution method to solve unsplittable MCNDP and its computational results are described. Finally, in section 4, the conclusion is presented.

2. Unsplittable MCND Problem

In unsplittable MCND problem, each commodity must follow exactly one rout between its origin and destination. It means, the flow variable must be a binary or 0-1 variable. Suppose a directed network $G = (N, A)$, in which N is the set of nodes and A is the set of directed arcs. Let K be the set of commodities, then the amount of each commodity k which must flow from its Origin $O(k)$ to its destination $D(k)$ is d_k. Let c_{ij}^{k} and f_{ij} be the per unit arc routing cost of commodity k on arc (i, j), and fixed arc designing cost of arc (i, j), respectively. The unsplittable MCND problem can be formulated as follows:

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^{k} x_{ij}^{k} + \sum_{(i,j) \in A} f_{ij}\, y_{ij} \qquad (1)$$

subject to

$$\sum_{j \in N} x_{ij}^{k} - \sum_{l \in N} x_{li}^{k} = \begin{cases} 1 & if \quad i = O(k) \\ -1 & if \quad i = D(k) \quad for\ all\ \ K \in k \\ 0 & otherwise \end{cases} \qquad (2)$$

$$\sum_{k \in K} x_{ij}^{k} d_{k} \le u_{ij} y_{ij} \qquad for\ all\ \ (i, j) \in A \qquad (3)$$

$$(x, y) \in S \qquad (4)$$

$$x_{ij}^k \; , \quad y_{ij} = 0 \; or \; 1 \qquad for \; all \; (i, j) \in A, \; k \in K \qquad (5)$$

Where, y_{ij} and x_{ij}^k are decision variables. y_{ij} is 0 if arc (i, j) is closed, and 1 if it is open. x_{ij}^k is 1 if commodity k is shipped on arc (i, j), otherwise is 0. The objective function (1), is to minimize variable and fixed costs. Constraints (2) are the usual balancing equations of network flow problem, which insure that each demand is starting from origin node, passing intermediate nodes, and reaching to destination node. Constraints (3) demonstrate that the sum of flows on each arc (i, j) must not exceed the capacity u_{ij} of the arc. S is the set of side constraint, which can be added to this general model.

These constraints might be for example the topological restrictions or multiple-choice restrictions of the arcs. Unsplittable MCND, which is presented above is arc-based formulation.

3. The Proposed SA for the Unsplittable MCND

The SA was proposed by [36]. The overall procedure of SA is as follows. The SA start with initial solution and move iteratively to neighborhood solutions, if the neighborhood solution is better than current solution, it set as a current solution, otherwise it accepted by the probability $\exp(-\Delta E / T)$ in which ΔE is the difference between current and neighborhood solutions, and T is a parameter, which called *temperature*. At each temperature, several iterations are executed and then temperature is gradually decreased. At the first step of the search, temperature is set to high value (more probability of accepting worst solutions), and gradually it is decreased to a little value at the end of the search (less probability of accepting worst solutions).

3.1. The Solution Representation

The SA needs a representation of solutions, which constructs a solution and the neighborhood structure is defined based on it. For the proposed SA method for unsplittable MCND problem, we inspired representation from [22] in which a representation for genetic algorithm for shortest path problem was proposed. To illustrate proposed neighborhood structure, suppose figure 1 as a partial network of unsplittable MCND problem with 3 commodities. The capacity of each arc has been shown on it. Table 1 shows origin, destination and demand of each commodity k. We use a $m \times n$ matrix to represent a solution, in which m and n is related to commodities and nodes, respectively. Each value (i, j) of this matrix shows a priority of node j for commodity i. For example, in figure 2a, the value of element $(1, 2)$ is 5,

that shows the priority of node 2 is 5 for commodity 1. To obtain a solution from this structure, we must decode each matrix value. To do this, we start from origin node in each commodity, and check its unvisited neighborhood, the neighborhood that has maximum priority must be the next node in the path. For example, in figure 2, for commodity 1, the priority of neighborhoods of node 1, i.e., nodes 2, 3, and 4 are 5, 3, and 9, respectively, so that the next node must be node 4.

This must be done for each next node in each commodity to reach destination node. By this procedure, the resulting solution has been shown in figure 2b.

The capacities and flows have been shown on each arc. This proposed representation, in some cases, may lead to paths that are not reached to their destinations. To handle this problem, in proposed algorithm, solution is checked and if this problem occurs, the related solution is eliminated and a new solution will be constructed.
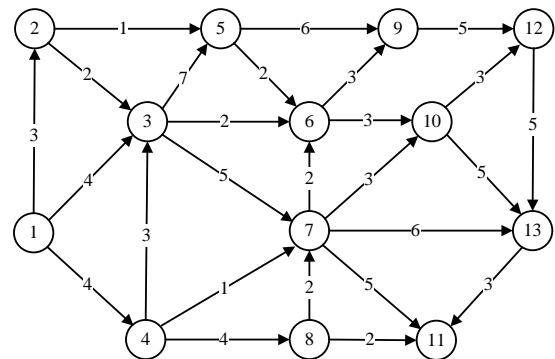

Fig. 1. A network example of MCNDP

Tab. 1. Origin, Destination and demand of each commodity k

	O(k)	D(k)	d_k
k_1	1	13	2
k_2	2	11	2
k_3	4	12	2

3.2. Neighborhood Structure

To move from a current solution to a neighborhood solution, we use swap in a fraction of commodities. To do this, a number of commodities are selected randomly for swap. This is a user-defined number, which can be seen as a parameter of the algorithm. In each selected commodity, two nodes are selected randomly for swap. A swap is done by changing the positions of priorities. This process has been shown in figure 3 in which commodity 1 is selected for swap. In this commodity,

nodes 2 and 11 are selected randomly to swap. Resulting path of each solution is shown below it.

In the special cases, these swaps might not lead to new solutions; i.e. there might be a situation that these swaps do not change a solution. The algorithm results show that in large networks with numerous arcs, this problem rarely occurs [21] and by selecting a fraction of commodities for swap instead of a single commodity there will be the low probability of occurrence this problem.
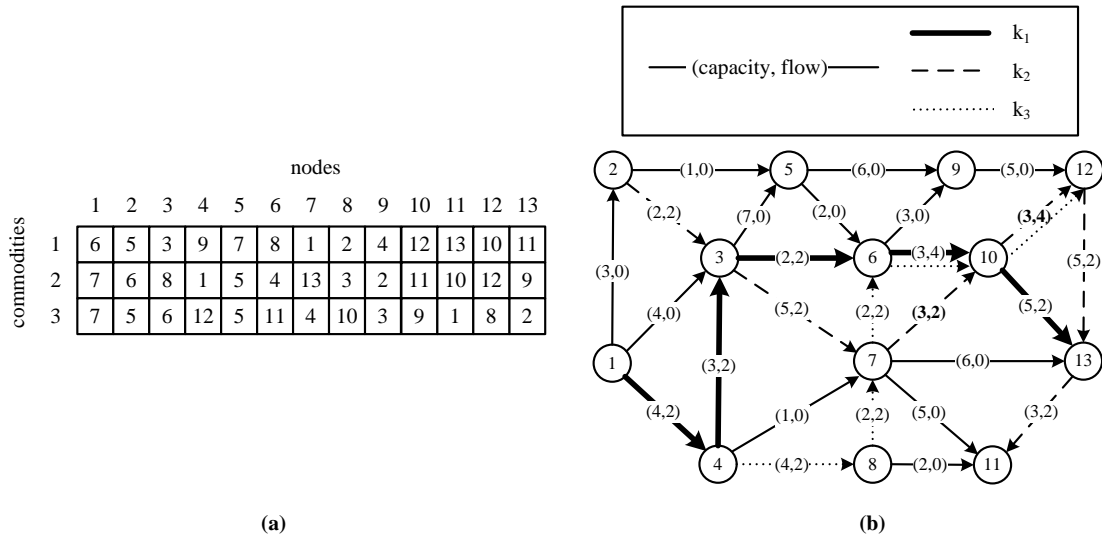

Fig. 2. (a) Representation of a solution (b) Resulting paths for commodities
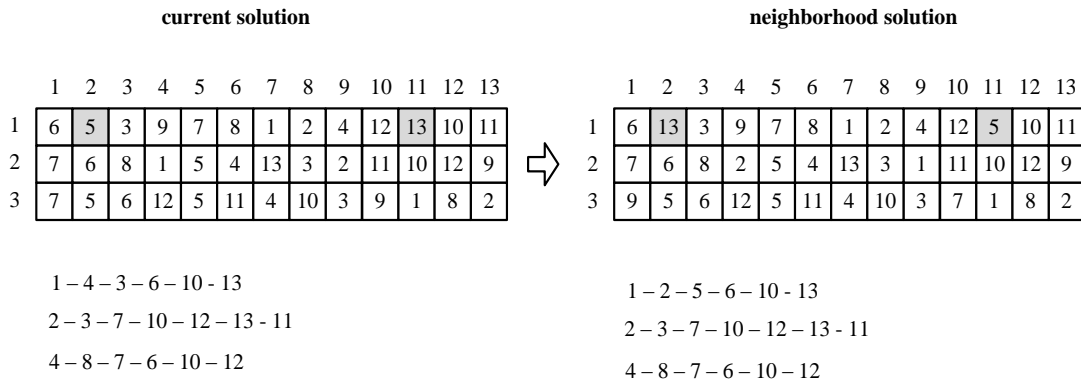

Fig. 3. Neighborhood structure for SA for network example of figure 1

3.3. Evaluating Objective Function and Handling Constraints

For evaluating objective function, the flows of each arc is calculated according to solution and fixed costs are added for opening arcs in addition to variable costs according to the flows. The main challenges of MCND problems are how to handle constraints. For unsplittable MCND problem, the sum of flows on each arc must not exceed the maximum capacity of them. To handle this constraint, after generating a solution, the sum of flows on each arc must be checked. If this value is more than maximum capacity, then a penalty value is added to objective value of the solution, so that the probabilities of acceptance these solutions are decreased. This penalty is added according to below equation:

$$(u_{ij} y_{ij} - \sum_{k \in K} x_{ij}^k d_k) w_p \ , \quad for \ (i, j) \in A \mid \sum_{k \in K} x_{ij}^k d_k > u_{ij} y_{ij}, \qquad (6)$$

where w_p is a user defined weight that represent penalties. This value can be seen as a parameter of the algorithm that must be tuned. This proposed penalty structure changes penalties according to solution. The worst the solution is, the more penalty is added.

For illustration, consider figure 2 in which the flow of arcs (6-10) and (10-12) exceed the maximum capacity of the arcs. Therefore, we must add a penalty to objective function of this solution.

3.4. The Overall Structure of the Proposed SA

Figure 4 shows a pseudocode for the proposed SA method for unsplittable MCND problem.

Input

$x = x_0$; /*create initial solution*/

$T = T_{max}$; /*set initial temperature*/

Repeat

Repeat /*in a constant temperature*/

Swap $(x = x')$; /*Generate a random neighbor*/

Calculate $f(x')$ and $f(x)$;

$\Delta E = f(x') - f(x)$

If $\Delta E <= 0$ **then** $x = x'$ /*accept neighborhood solution*/

Else accept x' with a probability $\exp(-\Delta E / T)$;

Until Equilibrium condition

$T = g(T)$; /*update temprature*/

Until Stopping criteria satisfied /*e.g. $T < T_{min}$*/

Output: Best solution found.

Fig. 4. Proposed SA pseudo code

Creating initial solution is done randomly. In swap procedure, current solution is moved to neighborhood solution according to swap structure we mentioned above (figure 2). To calculate $f(x')$ and $f(x)$, the related matrix must be decoded into solutions and then the objective function must be calculated. If the solution is infeasible according to capacity constraint, a penalty must be added to the objective function. Initial temperature, annealing function for updating temperature, and the other parameters are tuned in the next section.

3.5. Parameters Tuning

The parameters of the proposed algorithms are tuned using the Design of Experiments (DOE) approach and Design-Expert statistical software [51]. One can define an experiment as a test or series of tests in which purposeful changes are made to the input variables of a process or system so that we may observe and identify the reasons for changes that may be observed in the output response.

DOE refers to the process of planning the experiment so that appropriate data that can be analyzed by statistical methods will be collected, resulting in valid and objective conclusions [4].

The three basic principles of DOE are replication, randomization, and blocking. Replication means a repetition of the basic experiment. Replication has two important properties. First, it allows the experimenter to obtain an estimate of the experimental error. Second, if the sample mean is used to estimate the effect of a factor in the experiment, replication permits the experimenter to obtain a more precise estimate of this effect.

Randomization means that both the allocation of the experimental material and the order in which the individual runs or trials of the experiment are determined randomly. Randomization usually makes this assumption valid. Blocking is a design technique used to improve the precision with which comparisons among the factors of interest are made.

Often blocking is used to reduce or eliminate the variability transmitted from nuisance factors; that is, factors that may influence the experimental response but in which we are not directly interested [43]. The important parameters in DOE approach are response variable, factor, level, treatment and effect. The response variable is the measured variable of interest. In the analysis of metaheuristics, the typically measures are the solution quality and computation time [1]. A factor is an independent variable manipulated in an experiment because it is thought to affect one or more of the response variables.

The various values at which the factor is set are known as its levels. In metaheuristic performance analysis, the factors include both the metaheuristic tuning parameters and the most important problem characteristics [4]. A treatment is a specific combination of factor levels. The particular treatments will depend on the particular experiment design and on the ranges over which factors are varied. An effect is a change in the response variable due to a change in one or more factors [49].

Design of experiments is a tool that can be used to determine important parameters and interactions between them. Four-stages of DOE consist of screening and diagnosis of important factors, modeling, optimization and assessment. This methodology is called sequential experimentation, which is used to set the parameters in the DOE approach and has been used in this paper for the proposed algorithms [43].

Several samples with different characteristics are selected from OR-Library for parameter tuning and performance evaluation of this algorithm. These instances are randomly generated benchmark instances described in [18] and [19]. The same instances have been used in several other papers, such as [7], [10], [26], [11], [35], [46], and [39].

The proposed algorithm is implemented with Java programming language. Table 2 shows these samples and their characteristics. In the table, problems are sorted according to the number of commodities. In the two last columns, F means that the fixed costs are predominate relatively to the variables costs and V means the converse, while L means that the capacities are loose while T means that the capacities are tight.

Tab. 2. samples of OR-Library

Problem name	Number of nodes	Number of arcs	Number of Commodities	F/V	L/T
C35	20	230	40	V	T
C36	20	230	40	F	T
C41	20	300	40	V	L
C43	20	300	40	V	T
C44	20	300	40	F	T
C49	30	520	100	V	L
C50	30	520	100	F	L
C57	30	700	100	V	L
C58	30	700	100	F	L
C37	20	230	200	V	L
C38	20	230	200	F	L
C39	20	230	200	V	T
C40	20	230	200	F	T
C45	20	300	200	V	L
C46	20	300	200	F	L
C47	20	300	200	V	T
C53	30	520	400	V	L
C54	30	520	400	F	L
C64	30	700	400	F	T

To tune parameters for the proposed algorithm we classified instances into 3 classes, according to the number of commodities. The instances that have 40, 100 to 200, and 400 commodities are categorized in classes 1, 2, and 3, respectively. For the classes 1 to 3, instances C35, C58, and C64 are chosen as representatives for parameter tuning. Parameters tuning is done separately for each groups.

In the proposed algorithm, solution quality and CPU time are considered as the response variables. Factors, their levels, and the final obtained values are shown in Table 3. Each block is considered with 16 treatments and main effects. As a solution quality indicator, for each instance the average deviation from optimal value or lower bound result is calculated. Lower bound is used when CPLEX con not reach to optimal value.

$$relative\ gap = \frac{obtained\ solution - optimum(lower\ bound)}{optimum(lower\ bound)} \quad (7)$$

We use geometric cooling function for decreasing temperature:

$$T = \alpha T, \quad (8)$$

where T and α, are temperature and cooling rate, respectively. As illustrated in table 3, several α values

are examined for this function to reach the best value of it.

Tab. 3. Initial, ranges, and final values for representative samples

(a) Parameter tuning for the problems of class 1

Parameter	Lowest value	Highest value	Final Value
Initial temperature	80000	120000	100000
Final temperature	0.001	0.1	0.01
Thermodynamic condition	50	200	100
α	0.99	0.999	0.994
Fraction of commodities for swap	2	6	1
Wight of penalty	100	1500	1000

(b) Parameter tuning for the problems of class 2

Initial temperature	2000	6000	3000
Final temperature	0.001	0.1	0.001
Thermodynamic condition	50	200	100
α	0.99	0.999	0.999
Fraction of commodities for swap	2	6	4
Wight of penalty	100	400	100

(c) Parameter tuning for the problems of class 3

Initial temperature	1000	4000	1000
Final temperature	0.001	0.1	0.001
Thermodynamic condition	50	200	100
α	0.99	0.999	0.999
Fraction of commodities for swap	5	6	4
Wight of penalty	100	400	100

3.6. Experimental Results

As we mentioned in pervious section, three of samples are used for parameter tuning. The algorithm is implemented with tuned parameter, for 16 remaining samples. The results are summarized in table 4. In column "CPLEX time", "10h" means the CPLEX solver cannot reach to global optimum within 10 hours and CPLEX is stopped at this point.

Figure 5 compares computational time of CPLEX and the proposed algorithm. This shows a significant difference in computational time between CPLEX and the proposed algorithm. Figure 6 shows the relative gap in different samples.

This figure shows the gaps are increased as the size of samples increased. Figure 7 compares the objectives of CPLEX and SA. Figure 5 and 7 illustrate that SA can find near optimal solution in much less time.

Tab. 4. Results of implementing algorithm for samples

| Problem	CPLEX objective	CPLEX time	SA objective	SA time	Relative gap %
C36	668699	0.41 sec	670928	9 sec	0.33
C41	429335	0.4 sec	433929	10 sec	1.07
C43	501766	0.4 sec	511384	8 sec	1.92
C44	643395	0.55 sec	656412	9 sec	2.02
C49	54387	7 min 23 sec	57422	1 min 24 sec	5.58
C57	47883	9 min 10 sec	51505	3 min 57 sec	7.56
C58	61204	10 h	68179	8 min 15 sec	11.40
C37	94111	10 h	118785	9 min 16 sec	26.22
C39	99099	10 h	112792	13 min 24 sec	13.82
C40	138355	10 h	167006	8 min 54 sec	20.71
C45	76441	10 h	91702	8 min 15 sec	19.96
C46	119047	10 h	151513	8 min 40 sec	27.27
C47	77006	10 h	88550	8 min 28 sec	14.99
C53	115414	10 h	134587	21 min 34 sec	16.61
C64	138712	10 h	168070	20 min 14 sec	21.16
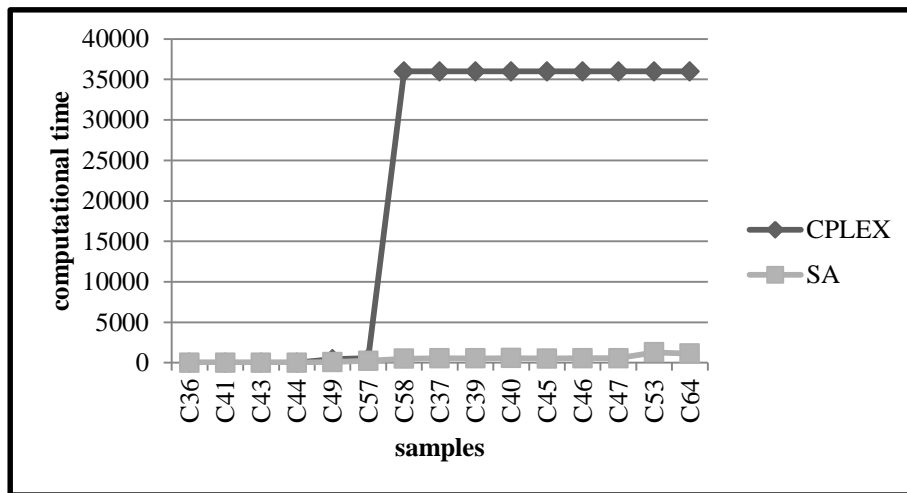
Fig. 5. Computational time for CPLEX and SA in different samples
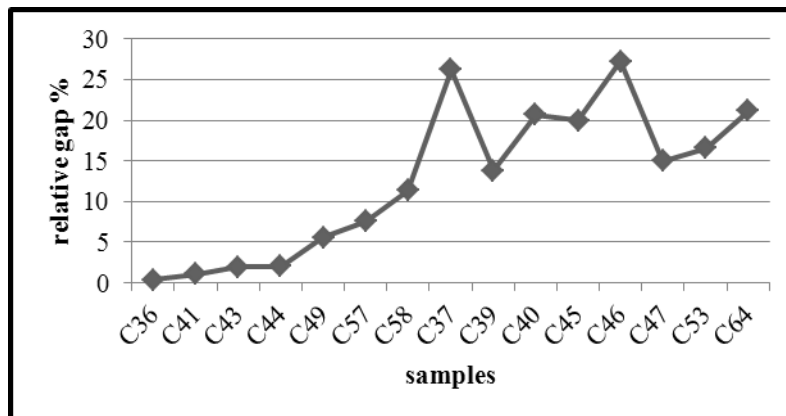

Fig. 6. Relative gap of SA in different samples
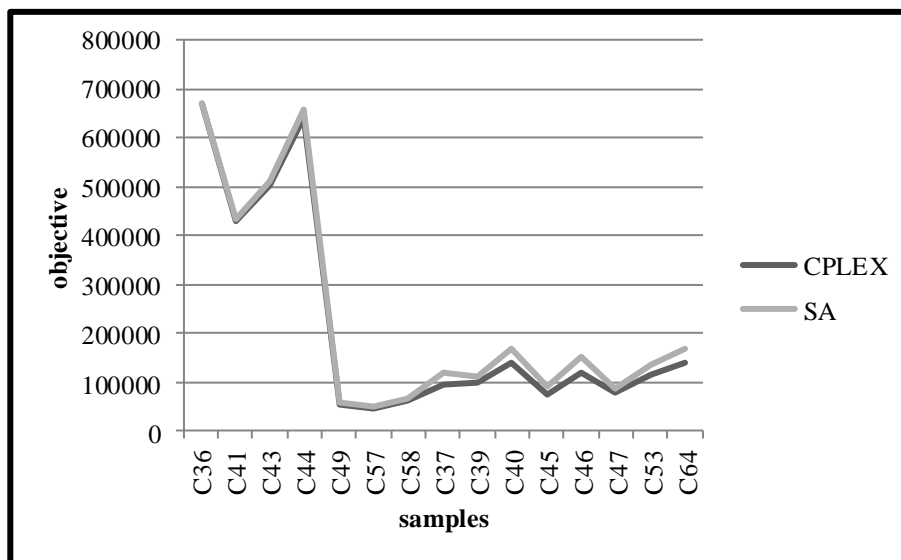
Fig. 7. Objectives of CPLEX and SA in different sample

4. Conclusions

In this paper, we have proposed a SA solution method with an innovative solution representation and neighborhood structure for unsplittable multicommodity capacitated network design problems. The results show that SA can find near optimal solution in much less time than exact algorithm. We hope proposed SA approach for unsplittable MCND problem will draw more researchers to develop and implement this approach for real-life problems. In addition, the proposed representation and neighborhood structure can be used in another metaheuristics such as genetic algorithm, and tabu search, so that, it can be another new search field.

References

[1] Adenso, B., Laguna, M., *"Fine-Tuning of Algorithms Using Fractional Experimental Designs and Local Search,"* Operations Research, Vol. 54, 2006, pp. 99–114.

[2] Balakrishnan, A., Magnanti, T.L., Mirchandani, P., *"Network Design,"* In: Dell'Amico, M., Maffioli, F., & Martello, S., (Eds.), Annotated Bibliographies in Combinatorial Optimization, John Wiley & Sons, 1997, pp. 311-329.

[3] Beasley, J.E., *"OR-Library: Distributing Test Problems by Electronic Mail,"* Journal of the Operational Research Society, Vol. 41, 1990, pp. 1069–1072, http://people.brunel.ac.uk/~mastjjb/jeb/info.html.

[4] Biratteri, M., Tuning Metaheuristics: A machine learning perspective, Springer, Heidelberg, 2009.

[5] Chouman, N., Crainic, T.G., *"A MIP-Tabu Search Hybrid Framework for Multicommodity Capacitated Fixed-charge Network Design,"* working paper, Centre de recherche sur les transports, Universite de Montreal, QC Canada, 2010.

[6] Crainic, T.G., Dejax, P., Delorme, L., *"Models for Multimode Location Problem with Interdepot Balancing Requirement,"* Annals of Operation Research, Vol. 18, 1989, pp. 277-302.

[7] Crainic, T.G., Gedreau, M., Farvolden, J.M., *"A Simplex-Based Tabu Search Method for Capacitated Network Design,"* INFORMS Journal on Computing, Vol. 12, 2000, pp. 223-236.

[8] Crainic, T.G., *"Service Network Design in Freight Transportation,"* European Journal of Operational Research, Vol. 122, 2000, pp. 272–288.

[9] Crainic, T.G., Frangioni, A., Gendron, B., *"Bundle-based Relaxation Methods for Multicommodity Capacitated Fixed Charge Network Design,"* Discrete Applied mathematics, 2000.

[10] Crainic, T.G., Gendron, B., Hernu, G., *"A Slope Scaling/Lagrangean Perturbation Heuristic with Long-Term Memory for Multicommodity Capacitated Fixed-Charge Network Design,"* Journal of Heuristics, Vol. 10, 2004, pp. 525-545.

[11] Crainic, T.G., Li, Y., & Toulouse, M., *"A First Multilevel Cooperative Algorithm for Capacitated Multicommodity Network Design,"* Computers & Operations Research, Vol. 33, 2006, pp. 2602–22.

[12] Doreo, J., Siarry, E., Petrowski, A., Taillard, E., Metaheuristics for Hard optimization, Springe, Heidelberg, 2006.

[13] Dorigo, M., Optimization, Learning and Natural Algorithms, PhD thesis, Politecnico di Milano, Italy, 1992.

[14] Dorigo, M, Stützle, T., Ant Colony Optimization, MIT Press, Cambridge, 2004.

[15] Eiben, A.E., Smith J.E., Introduction to Evolutionary Computing. Springer, Amsterdam, 2003.

[16] Feterolf, P., Anandanlingham, G., *"Optimizing Interconnections of Local Area Networks: An Approach using Simulated Annealing,"* ORSA Journal of Computing Vol. 3, 1991, pp. 275–287.

[17] Frangioni, A., Gendron, B., *"0-1 Reformulation of the Multicommodity Capacitated Network Design Problem,"* Discrete Applied Mathematics, Vol. 157, 2008, pp. 1229-1241.

[18] Gendron, B., Crainic, T.G., *"Relaxations for Multicommodity Capacitated Network Design Problems,"* Publication CRT-945, Centre de recherche sur les transports, Université de Montréal, 1994.

[19] Gendron, B., Crainic, T.G., *"Bounding Procedures for Multicommodity Capacitated Fixed Charge Network Design Problems,"* Publication CRT-96-06, Centre de recherche sur les transports, Université de Montréal, 1996.

[20] Gendron, B., Crainic, T.G., Frangioni, A., *"Multicommodity Capacitated Network Design,"* In: Sanso, B., & Soriono, P., (Eds.), Telecommunication Network Planning, Kluwer, Norwell, 1998, pp. 1-19.

[21] Gen, M., Cheng, R., Genetic Algorithms and Engineering Design, John Wiley & sons, New York, 1997.

[22] Gen, M, Cheng R, & Wang D. Genetic Algorithm for Solving Shortest Path Problems, proceeding IEEE international conference on evolutionary computation, Indianpolis, Indiana; 1997, pp. 401-406.

[23] Gen, M., Cheng, R., Genetic Algorithms and Engineering Optimization, John Wiley & sons, New York, 2000.

[24] Gen, M., Cheng, R., Oren, Sh.S., *"Network design Techniques Using Adapted Genetic Algorithms,"* Advance in Engineering Software, Vol. 32, 2000, pp.731-744.

[25] Gen, M., Kumar, A., Kim, J.R., *"Recent Network Design Techniques Using Evolutionary Algorithms,"* International Journal of Production Economics, Vol. 98, 2005, pp. 251-261.

[26] Ghamlouche, I., Crainic, T.G., & Gendreau, M., *"Cycle-Based Neighbourhoods for Fixed-Charge Capacitated Multicommodity Network Design,"* Operations Research, Vol. 51, 2003, pp. 655 – 667.

[27] Glover, F., *"Future Paths for Integer Programming and Links to Artificial Intelligence,"* Computers and Operations Research, Vol. 13, 1986, pp. 533–549.

[28] Glover, F., & Laguna, M., Tabu Search, Kluwer, London, 1997.

[29] Glover, F., & Kochenberger, G.A., Handbook of Metaheuristics, Kluwer, Norwell, 2003.

[30] Goldberg, D.E., Genetic Algorithms in Search, Optimization and Machine Learning, Addison-wesley, 1989.

[31] Hoff, A., Lium, A.G., Lokketangen, A., Crainic, T.G., *"A Metaheuristic for Stochastic Service Network Design,"* Journal of Heuristics, Vol. 16, 2010, pp. 653-679.

[32] Holland, J.H., Adaptation in Natural and Artificial Systems, University of Michigan Press: Ann Arbor, MI, 1975.

[33] Jayaraman, V., Ross, A., *"A simulated annealing methodology to distribution network design and management,"* European Journal of Operational Research, Vol. 144, 2002, pp. 629-645.

[34] Jin, Q., Li-xin, M., *"Combined Simulated Annealing Algorithm for Logistics Network Design Problem,"* ISA 2009 International Workshop on Intelligent Systems and Applications, IEEE, 2009, pp. 1-4.

[35] Katayama, N., Chen, M., & Kubo, M., *"Capacity Scaling Heuristic for the Multicommodity Capacitated Network Design Problem,"* Journal of Computational and Applied Mathematics, Vol. 232, 2009, pp. 90-101.

[36] Kirkpatrick, S., Gelatt, C., Vecchi, M., *"Optimization by Simulated Annealing,"* Science, Vol. 220, 1983, pp. 671-680.

[37] Kleeman, M.P., Lamont, G.B., Hopkinson, K.M., Graham, S.R., *"Solving Multicommodity Capacitated Network Design Problems using a Multiobjective Evolutionary Algorithm,"* Proceedings of IEEE Symposium on Computational Intelligence in Security and Defense Applications, 2007.

[38] Magnanti, T.L., Wong, R.T., *"Network Design and Transportation Planning: Models and Algorithms,"* Transportation Science, Vol. 18, 1984, pp. 1-55.

[39] Martín, I.R., González, J.J.S., *"A Local Branching Heuristic for the Capacitated Fixed-Charge Network Design Problem,"* Computers & Operations Research, Vol. 37, 2010, pp. 575-581.

[40] Minoux, M., *"Network Synthesis and Optimum Network Design Problems: Models, Solution Methods and Applications,"* Network, Vol. 19, 1989, pp. 313-360.

[41] Minoux, M., *"Discrete Cost Multicommodity Network Optimization Problems and Exact Solution Methods,"* Annals of Operations Research, Vol. 106, 2001, pp. 19-46.

[42] Minoux, M., *"Optimum Network Design Models and Algorithms in Transportation and Communication,"* International Journal of Logistic: Research and Application, Vol. 6, 2003, pp. 5-15.

[43] Montgomery, D., Design and Analysis of Experiments, John Wiley and Sons, 2005.

[44] Morago, R.J., DePuy, G.W., Whitehouse G.E., *"A Solution Methodology for Optimization Problems,"* In: Badiru, A.B., (Ed.), Metaheuristics, Taylor & Francis Group, 2006, pp. 1011-1013.

[45] Oimoen, S.C., Peterson, G.L., Hopkinson, K.M., *"Network Formation Using Ant Colony Optimization,"*, Lecture Notes in Computer Science, Vol. 5217, 2008, pp. 405-406.

[46] Pedersen, M.B., Crainic, T.G., *"Madsen O.B.G. Models and Tabu Search Metaheuristics for Service Network Design with Asset-Balance Requirements,"* Transportation Science, Vol. 43, 2009, pp. 158-177.

[47] Pishvaee, M.S., Kianfar, K., Karimi, B., *"Reverse Logistic Network Design using Simulated Annealing,"* The International Journal of Advanced Manufacturing Technology, Vol. 47, 2009, pp. 269-281.

[48] Poorzahedy, H., Rouhani, O., *"Hybrid Metaheuristic Algorithms for Solving Network Design Problems,"* European Journal of Operational Research, Vol. 182, 2006, pp. 578-596.

[49] Ridge, E., *"Design of Experiments for the Tuning of Optimization Algorithms"*, PhD thesis, Department of Computer Science, University of York, United Kingdom, 2007.

[50] Talbi, E., Metaheuristics: From Design to Implementation, John Wiley & sons, Hoboken, 2009.

[51] Vaughn, N., Polnaszek, C., Smith, B., Helseth, T., Design-Expert 6 User's Guide, Stat-Ease Inc, 2000.

[52] Vernekar, A., Anandanlingham, G., Dorny, C., *"Optimization of Resource Locations in Hierarchical Computer Networks,"* Computers and Operations Research, Vol. 17, 1990, pp. 375–388.

[53] Zaleta N.C., Socarras, A.M.A., *"Tabu Search-based Algorithm for Capacitated Multicommodity Network Design Problem,"* Proceedings of the 14th International Conference on Electronics, Communications and Computers, IEEE, 2004.