Computational Economics (2006) 27: 273–293
DOI: 10.1007/s10614-006-9028-4

An Enhanced Dynamic Slope Scaling Procedure with Tabu Scheme for Fixed Charge Network Flow Problems

DUKWON KIM[1], XINYAN PAN[1] and PANOS M. PARDALOS[2,*]
[1]*Industrial and Information Engineering, University of Tennessee, Knoxville, TN 37996*
[2]*Center for Applied Optimization, Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611*

Abstract. A heuristic algorithm for solving large scale fixed charge network flow problems (FCNFP) based on the dynamic slope scaling procedure (DSSP) and tabu search strategies is presented. The proposed heuristic integrates the DSSP with short-term memory intensification and long-term memory diversification mechanisms in the tabu scheme to improve the performance of the pure DSSP. With the feature of dynamically evolving memory, the enhanced DSSP evaluates the solutions in the search history and iteratively adjusts the linear factors in the linear approximation of the FCNFP to produce promising search neighborhoods for good quality solutions. The comprehensive numerical experiments on various test problems ranging from sparse to dense network structures are reported. The overall comparison of the pure DSSP, the enhanced DSSP, and branch and bound (B&B by cutting-edge MIP optimizer in CPLEX) is shown in terms of solution quality and CPU time. The results show that the enhanced DSSP approach has a higher solution quality than the pure DSSP for larger scale problems.

Key words: dynamic slope scaling procedure, fixed charge network flow problems, tabu search

1. Introduction

The fixed charged network flow problem (FCNFP) is one of the practical branch problems in the minimum cost network flow problems. Besides the variable cost on each arc in the network, a fixed setup cost occurs when there is a flow on an arc. The fixed setup costs are very common in real world applications, which include handling fees, changeover times, charter rental and docking fees, etc. The central decision "to go or not to go" can be modeled by imposing fixed charges on the arcs of a network. FCNFP is known as NP-hard and usually formulated as a mixed-integer programming (MIP) model. The solution time of MIP problems with exact solution methods, typically the branch and bound method (B&B), increases exponentially as the problem size increases. In real life, it is not practical for a firm to solve

*Research has been partially supported by NSF and CRDF grants.

the large-scale FCNFP problems with the branch and bound techniques because the search process may last several days. Hence, we are interested in developing heuristics to achieve a reasonable trade-off between the proximity of the solution and the computational time.

Exact solution approaches based on the B&B and cutting plane methods were developed by Gray (1971), Steinberg (1970), Palekar et al. (1990), Hochbaum (1989), Barr et al. (1981), Suhl (1985) and Cabot and Erenguc (1984). However, the enumerative B&B method needs several hundreds of branching steps and cuts even for a 4 by 4 problem. The exponentially increasing number of subproblems that need to be solved in order to find the optimal solutions of FCNFP makes B&B very unwieldy in large-scale FCNFP. Typically, tremendous computational time is required to justify the optimality of the current solution. By contrast, heuristic approaches have some merits on these aspects. Several heuristic approaches had been proposed in the past three decades.

To date, quite a few of the heuristic approaches in the fixed charge problems are based on Lagrangean relaxation and decomposition. These types of approaches were presented by Geoffrion (1974), Magnanti et al. (1986), Gendron and Grainic (1994) and Crainic and Frangioni (1994). Gendron and Grainic (1994) presented a methodology that uses a series of Lagrangian relaxations in the design of the multi-commodity capacitated fixed charge network problems.

The application of the tabu search (Glover and Laguna (1989, 1990, 1997) in the Fixed Charge Transportation Problem (FCTP) and FCNFP was proposed by Sun et al. (1998) and Crainic and Farvolden (2000). Sun et al. introduced an effective and thorough tabu search algorithm in the uncapacitated FCTP with immediate memory to find the local approximated optima, short-term memory to intensify the optimal search, and long-term memory to diversify the search among the least visited arcs. Glover (1994) proposed an approach called the ghost image process for the fixed charge problem with the parametric deformation. It progressively improves the solution by certain method (the tabu search was recommended in his paper).

Kim and Pardalos (1999) developed a new heuristic framework, called Dynamic Slope Scaling Procedure (DSSP), to solve the capacitated FCNFP problems. DSSP parameterizes a nonlinear fixed charge factor into a linear factor associated with each arc, and adjusts the linear factors iteratively to reflect the exact original variable and the fixed cost in order to approximate the true objective function value.

In this paper, we propose a heuristic that combines DSSP with the short-term and long-term memory techniques from the tabu search, which we refer as the enhanced DSSP hereafter. In addition to the computational experiment on DSSP by Kim and Padarlos (1999), we have further examined the performance of DSSP on larger scale FCNFP. The paper is organized as follows. Section 2 presents a mathematical formulation of FCNFP. Section 3 includes a brief description of DSSP approach. Section 4 includes a detailed design of the enhanced DSSP procedure, including the description of the tabu strategies and overall mechanism for the enhanced DSSP. The description of generating test problems and the comparisons of the computational results for DSSP, the enhanced DSSP and the branch and bound method are given in Section 5.

2. Problem Description

Given a directed graph $G = (N, A)$ consisting of a set N of n nodes and a set of A of m arcs, let x_{ij} denote the flow on the arc from node i to node j, c_{ij} as the variable cost and s_{ij} the fixed cost. Let u_{ij} denote the capacity of arc (i, j), and b_i the amount to be sent from supply and received at demand node. "O" represents the source node and "D" represents the demand node. The general problem is given as:

$$\min f(x) = \sum_{(i,j)\in A} f_{ij}(x_{ij}) \tag{1}$$

$$s.t \sum_{j=1}^{n} x_{ij} - \sum_{k=1}^{n} x_{ki} = \begin{Bmatrix} b_i & \text{if} & i = O \\ -b_i & \text{if} & i = D \\ 0 & & \text{otherwise} \end{Bmatrix} \quad \text{for} \quad i = 1, 2, \ldots, n \tag{2}$$

$$\begin{aligned} x_{ij} &\le u_{ij} \quad \text{for} \quad (i, j) \in A \\ x_{ij} &\ge 0 \quad \text{for} \quad (i, j) \in A \end{aligned} \tag{3}$$

where f is discontinuous at zero and separable such that for each arc (i, j), f_{ij} has a form:

$$f_{ij}(x_{ij}) = \begin{cases} 0 & \text{if } x_{ij} = 0 \\ s_{ij} + c_{ij}x_{ij}, \quad s_{ij} \ge 0 & \text{if } x_{ij} > 0. \end{cases} \tag{4}$$

3. DSSP

The dynamic slope scaling procedure (DSSP) is developed by Kim and Pardalos (1999). The computational merit of this approach is to approximate a solution for FCNFP by solving successive LP problems with recursively updated coefficients of the objective function. At each iteration, the linear factor is adjusted to reflect both the current variable cost and the fixed cost, while the overall marginal cost depends on the current level of each activity. Initially, when there is no flow through the arcs, a non-linear cost of an arc is transformed into a linear factor given as:

$$\bar{c}_{ij}^{0} = c_{ij} + s_{ij}/u_{ij} \quad \text{for } i, j = 1, 2, \ldots, n\ . \tag{5}$$

The objective function of FCNFP is then approximated as a linearized problem:

$$\min f(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} \bar{c}_{ij}x_{ij} \tag{6}$$

$$s.t \sum_{j=1}^{n} x_{ij} - \sum_{k=1}^{n} x_{ki} = \left\{\begin{array}{lll} b_i & \text{if} & i = O \\ -b_i & \text{if} & i = D \\ 0 & & \text{otherwise} \end{array}\right\} \quad \text{for } i = 1, 2, \ldots, n \tag{7}$$

$$\begin{array}{lll} x_{ij} \le u_{ij} & \text{for} & (i, j) \in A \\ x_{ij} \ge 0 & \text{for} & (i, j) \in A. \end{array} \tag{8}$$

Note that the constraints of the FCNFP are not changed in this linear approximation. At each iteration, a new linear approximation is solved with adjusted linear factors. The linear factor $\bar{c}_{ij}$ in the next linear approximation is updated with x_{ij} in the current solution. If this x_{ij} does not justify the fixed cost on arc (i, j), the LP solving mechanism will either increase the value of x_{ij} to justify the investment or drop it to zero in the next iteration. This means that marginal fixed cost, s_{ij}/x_{ij} decreases as x_{ij} increases, so that the value of linear factor (i.e. cost coefficient in the linear approximation) will decrease to keep x_{ij} as an active variable. Kim and Pardalos [13] proposed two updating schemes with different strategies in setting $\bar{c}^k_{ij}$ when $x^{k-1}_{ij} = 0$. The updating schemes are given as follows:

$$\bar{c}^k_{ij} = \begin{cases} c_{ij} + s_{ij}\big/x^{k-1}_{ij} & \text{if } x^{k-1}_{ij} > 0 \\ \max_{1 \le l \le k}\left\{\bar{c}^l_{ij} | x^{l-1}_{ij} > 0\right\} & \text{if } x^{k-1}_{ij} = 0 \end{cases} \tag{9}$$

or

$$\bar{c}^k_{ij} = \begin{cases} c_{ij} + s_{ij}\big/x^{k-1}_{ij} & \text{if } x^{k-1}_{ij} > 0 \\ \bar{c}^r_{ij} & \text{if } x^{k-1}_{ij} = 0 \end{cases} \tag{10}$$

where $\max_{1 \le l \le k}\{\bar{c}^l_{ij} \,|\, x^{l-1}_{ij} > 0\}$ is the maximum linear factor of the arc (i, j) in the iteration history with non-zero flow in the previous iteration; $\bar{c}^r_{ij}$ is the most recent $\bar{c}_{ij}$ in the iteration history for arc (i, j) with non-zero flow in the previous iteration. DSSP continues until identical solutions are obtained in two consecutive iterations. The numerical experiment of Kim and Pardalos (1999) has shown that for the large-scale problems, the scheme (10) has a slightly better performance than (9). Thus, we choose (10) as an updating scheme in the DSSP phase in the enhanced DSSP.

According to the computational experiment conducted by Kim and Pardalos (1999), DSSP has quite satisfactory performance in solving the small and medium size FCNFP problems, and finding exact solutions in most cases of the small size problems. However, in the medium and large-scale problems, it is true that the chance of achieving the true optima by DSSP becomes scarcer. In such cases, DSSP stops at a local optimum which is still far away from the exact solution. In order to improve the performance of DSSP, we propose a heuristic that combines DSSP with the tabu scheme to force DSSP to continue the additional search when DSSP fails to make any progress.

4. Enhanced DSSP with Tabu Search

In the proposed algorithm, the tabu search characterizes subsets of potential moves and tabus, according to the past search records to either continue the search near or divert the search from the already visited feasible region. The two classical components of the tabu search, the intensification and diversification strategies, are used in the enhanced DSSP. The frequency records the number of iterations an arc appeared in the earlier solutions. Two types of frequency memory are used in the algorithm: (1) frequency memory throughout the whole search process, which is also referred to as the long-term memory; (2) frequency memory in one separate DSSP phase, which is, along with the quality memory, also referred to as the short-term memory. The recency-based memory keeps track of the arc being basic or non-basic in the recent history of solutions. The arcs being active for certain iterations of consecutive intensification and diversification processes are labeled as *tabu-active* and will not be selected into the candidate list in the following intensification and diversification processes. This prevents the solutions containing *tabu-active* arcs from belonging to the modified neighborhood and hence from being revisited.

In general, the enhanced DSSP with the tabu scheme consists of three phases: (1) DSSP to solve the linear approximation of FCNFP problems, (2) the intensification process based on the short-term memory (3) the diversification process based on the long-term memory. The enhanced DSSP starts DSSP phase with the same initial scheme shown in (5). Except that it records the performance data during DSSP iterations, the first DSSP phase is very similar to the pure DSSP. The intensification process then is invoked to apply the initial linearization scheme and lead a new iteration of DSSP phase to further search the neighborhood in which the current best solution occurred. After the intensification iterations repeated for a certain period of time (the number of the intensification iterations will be discussed in Section 4.2), the search process moves on to the diversification iteration to explore a never visited region for better solutions. Since each diversification process generates a new search region that has not been explored before, it is necessary for the intensification iterations to further examine this region after diversification iteration, especially if it has identified a solution that is better than the current best solution. Therefore, the whole search process can be interpreted as: DSSP phase is the actual search executor, while the intensification and diversification processes define the strategic directions that the search executor should follow.

4.1. TABU SEARCH STRATEGIES

We utilize the long-term and short-term memory in designing MOVE sets, tabu set, aspiration set, rules of managing these sets and the initial solution schemes in the intensification and diversification processes.

4.1.1. *MOVE Sets*

The MOVE sets are used to store the arcs that will be encouraged into the basis in the following search process. After the first DSSP phase, a different initial linearization scheme than (5) will impose incentives to the arcs in the MOVE sets to let these arcs enter the basis in the next round of DSSP phase, which drive the following DSSP to search a modified neighborhood. This may lead to a result that improves the current best solution so far, which we refer as an "improved-best solution" in the procedure. Hereafter, we call the arcs in the MOVE sets as the "trigger arcs" because they play a critical role in distinguishing the new search neighborhood from the previous one. We designed two MOVE sets α and β for the intensification process and diversification process respectively.

The construction of MOVE set α in each intensification process is based on the short-term memory of frequency, recency and quality. The frequency memory records arcs frequently visited in DSSP phase immediately before the intensification process. The quality memory records the specific neighborhood where the improved-best solution is found. The recency memory prevents the same solution from being revisited. Thus, the intensification process defines α in a way to lead the search process to further explore the immediate neighborhoods near the improved-best solution. The details of selection criteria and incentive applied to the trigger arcs in the intensification process will be discussed in Section 4.2.

The construction of MOVE set β in each diversification process is based on the recency memory and the long-term memory of frequency. More precisely, the frequency memory here records those arcs that have never been visited throughout the entire search history. By selecting and encouraging never visited arcs into the basis in the following search process, the diversification process, in fact, directs the search process into a not-yet-visited neighborhood. The recency memory records the tabu status of the arcs. Note that after the diversification iteration, the intensification process will be invoked again to direct another round of DSSP phase to either further search the newly defined neighborhood if an improved-best solution is found in the previous diversification iteration, or return to the neighborhood in which the most recent improved-best solution was found. The details of selection criteria and incentive to the trigger arcs in the diversification process will be discussed in Section 4.3.

4.1.2. *Tabu Set*
The tabu set τ stores the arcs that have been moved for a certain period of time, denoted as tm_max_sh and tm_max_lm for intensification and diversification processes, respectively. It may cause DSSP to return to the recently visited solution. These arcs become tabu-active and are kept from being selected into α or β. Let tb_{ij} denote the period of time that arc (i, j) has been a tabu. When arc (i, j) sits in τ for a certain period of time, denoted as tb_max_sh and tb_max_lm for intensification and diversification processes, respectively, it is allowed to enter the MOVE sets again for the succeeding intensification and diversification until it reaches tm_max_sh or tm_max_lm to be forbidden again. Note that the tabu restriction is only applied to the arcs selected into α or β based on the frequency memory. The arcs selected by aspiration criterion are exempted from the tabu status examination.

4.1.3. *Aspiration Criterion*
The aspiration criterion is used in the intensification process to ensure the neighborhoods around the improved-best solution, are thoroughly searched. When the improved-best solution occurs in DSSP phase, the arcs with the most dramatic flow change from the previous DSSP iteration (e.g. changing from non-basic to basic) have the main contribution in finding the improved-best solution. These arcs are then kept in the inspiration set ζ during DSSP phase and will be selected into α regardless of their tabu status in the following intensification process. Note that the linear factors associated with the improved-best solution are also recorded and will be used in the initial linearization scheme in the following intensification process. This will be discussed in greater detail in Section 4.2.

4.2. INTENSIFICATION PROCESS WITH SHORT-TERM MEMORY

The intensification process is to direct DSSP phase search more thoroughly in the modified neighborhood in which an improved-best solution is found. The intensification process selects appropriate candidates into the MOVE set α and then, differentiates the initial linearization schemes for these trigger arcs and the other ones for DSSP phase following immediately. The intensification only uses the performance data collected from DSSP phase immediately before it. This immediate-before DSSP phase keeps track of the number of times that the arc is active for all the arcs, denoted as $t\_xnonzero\_sh_{ij}$ and memorizes the arcs with drastic changes in the improved-best solution from DSSP phase with the aspiration set, denoted as ζ, and the linear factors associated with the most recent improved-best solution for the rest arcs, denoted as $\bar{c}^b_{ij}$. The set ζ then forms a part of the MOVE set α. The rest of arcs in α are stored in the short-term memory and not *tabu-active*. As a result, the intensification process selects the arc into α if it satisfies either of the following conditions:

$$\{(i, j) \mid t\_xnonzero\_sh_{ij} \geq k/2, (i, j) \notin \tau\} \text{ or}$$
$$\{(i, j) \mid (i, j) \in \zeta\}$$

where k is the number of iterations in the DSSP phase immediately before the intensification. We consider the arcs with $t\_xnonzero\_sh_{ij} \geq k/2$ are virtually frequently visited arcs. This will be further explained in Section 5.2.

For the arcs, which do not belong to α, we set their linear factors to $\bar{c}_{ij}^{b}$, and for the arcs in α, only variable costs are included in their linear factors, significantly less than the linear factors subject to the fixed charges. The initial setting scheme of the linear factors for a new round of DSSP phase immediately after the intensification process is given as follows:

$$\bar{c}_{ij}^{0} \begin{cases} c_{ij} & \text{if } (i,j) \in \alpha \\ \bar{c}_{ij}^{b} & \text{otherwise.} \end{cases} \tag{11}$$

The recency memory, the frequency memory and the quality memory collaboratively generate a modified neighborhood that is close to, but slightly different than the one containing the current improved-best solution. This leads the following DSSP phase to further search the good solution areas. The iteratively updated intensification process upgrades neighborhoods that are close to the most recent improved-best solution for DSSP phase in search for better solutions.

Since the process memorizes the arcs at the aspiration level in the inspiration set ζ until a new improved-best solution is found, the search around the same improved-best solution neighborhood may be continued in the intensification iterations that filter through several diversification iterations if no new improved-best solution is found during these diversification process. In this case, the search near the same improved-best solution from the earlier intensification process is resumed after the diversification. Even though the diversification process may have changed the set of frequently visited arcs to be selected in the following intensification, the number of these arcs only accounts for a very small portion of the total number of arcs in the problem, therefore, with most linear factors still set to the same $\bar{c}_{ij}^{b}$, the search process, in fact, returns to the improved-best solution neighborhood. As a result, it seems sufficient to set the number of intensifications to 1 within a single diversification iteration. Moreover, according to our observation in the computational experiment, the number of intensification iterations has no significant impact on the performance of the enhanced DSSP. The selection of tb_max_sh and tm_max_sh is empirical and will be discussed in Section 5.2.

4.3. DIVERSIFICATION PROCESS WITH LONG-TERM MEMORY

The diversification process is designed to lead the algorithm to divert the search process into a new region that has not yet been explored. The diversification process selects a certain number of unvisited arcs into MOVE set β of the diversification and linearizes the initial linear factors to direct DSSP phase following the diversification to explore a new unexplored neighborhood. Four main factors for an effective diversification mechanism are the size of MOVE set β, denoted as *divno*, the selection criteria of trigger arcs, the initial linearization scheme for the following DSSP phase, and the period of time an arc being a move or a tabu.

It is not surprising that there are many more arcs left unvisited than visited in the pure DSSP or in a single DSSP phase in the enhanced DSSP. Obviously, simply selecting all the inactive arcs into β will not produce a good direction for the search process to continue searching for better solutions. Therefore, unlike the construction of the MOVE set α in the intensification process, that selects all the frequently visited arcs in the short-term memory as a part of α, the construction of β depends on how many inactive arcs should be selected and under what extent of inactiveness the arc can be selected into β. Therefore, the construction of β becomes more empirical. However, too small size of MOVE set β will not produce sufficient influence and will be dominated by the influence of the other arcs outside of β, which results in the process still being trapped at a local optimum. On the other hand, selecting a large β may include too many always-performing-bad arcs, which will tremendously dilute the effectiveness of the good candidates and lead the search process into unproductive neighborhoods. We will discuss different settings of these two factors in Section 5.2.

In order to decide which arcs should be selected into β, we adopt the concept of reduced cost. Let π denote the dual variables associated with constraints (2) and γ with constraints (3), respectively ($\pi, \gamma \geq 0$). The reduced cost associated with arc (i, j) in the linear approximation of FCNFP is then given by:

$$rc_{ij} = \bar{c}_{ij} + \Delta_{ij} = \bar{c}_{ij} + \pi_i - \pi_j + \gamma_{ij} \quad \forall (i, j) \in A. \tag{12}$$

The reduced cost indicates how strongly the associated arc tends to be active. The value of reduced cost is the amount, by which the coefficient of the associated variable needs to be improved so that the associated variable can enter the basis in the following LP iterations. In other words, the less the reduced cost is, the less cost we need to pay to bring this arc into the basis. The less cost means a less negative impact on the objective value. On the other hand, some other arcs in the changed basis may bring profits to the objective value (i.e. decreasing the objective value in minimization), which may offset and outstrip that negative impact caused by trigger arcs and hence result in a better solution. Therefore, it seems reasonable to choose the arcs with the least reduced cost into β.

While memorizing short-term performance information for the intensification process, DSSP phase also memorizes performance information throughout the whole search history, including the number of iterations the arc is active, denoted as $t\_xnonzero\_lm_{ij}$, the average reduced cost, denoted as avg_rc_{ij} and the average linear factors of each arc, denoted as $avg\_\bar{c}_{ij}$. Thus, the arcs that have never been visited can be defined as:

$$\eta = \{(i, j) \mid t\_xnonzero\_lm_{ij} = 0\}. \tag{13}$$

The diversification process selects from η *divno* number of arcs that are not tabu active and with the least reduced cost into β, which is given as:

$$\beta = \left\{(i, j) \mid \min^{(divno)}(avg\_rc_{ij}, (i, j) \in \eta \quad \text{and } (i, j) \notin \tau)\right\}. \tag{14}$$

Based on the reduced cost, the linear factors of the trigger arcs are then set to the average linear factor throughout the preceding iterations, denoted as $avg\_\bar{c}_{ij}$, subtracted by the average reduced cost, while the linear factors of the rest arcs are set to the average linear factor of the whole search history. Therefore, the initial linearization scheme for a new round of DSSP phase immediately after the diversification is given as follows:

$$\bar{c}^0_{ij} = \begin{cases} \text{avg\_}\bar{c}_{ij} - \text{avg\_}rc_{ij} & \text{if } (i, j) \in \beta \\ \text{avg\_}\bar{c}_{ij} & \text{otherwise.} \end{cases} \tag{15}$$

The computational results show that this incentive imposed on the arcs in β is quite sufficient to make the most inactive arcs enter the basis in the following DSSP phase. It seems that setting the maximum number of the diversification iterations to 20 empirically is sufficient for all tested problems.

Taking into account the fact that there are many more non-basic variables than basic variables in a basic feasible solution, we implement the tabu structure so that the period of time the arcs are eligible to be selected into candidate list β is usually several times longer than the period of time the arcs are forbidden in the diversification process (i.e. $tm\_max\_lm < tb\_max\_lm$). Yet the selection of tm_max_lm and tb_max_lm is empirical. We will discuss this in greater detail in Section 5.2.

4.4. OVERALL PROCEDURES

Phase 1: Initialization

Let Z denote the best objective value obtained through the iterations to date, which, in its turn, is the best solution the heuristic finds in the end, and let x_{ij} denote the variable value corresponding to Z. Let Z_local denote the objective value of the current DSSP iteration and x_local_{ij} denote the variable value corresponding to Z_local. Let $xpre_{ij}$ denote the variable value found in the previous DSSP iteration. Initially set $Z = \infty$, $x_{ij} = 0$ and $t\_xnonzero\_lm_{ij} = 0$, $tm_{ij} = 0$, $tb_{ij} = 0$. Initiate FCNFP problem by transforming it into the linear approximation as described in (5).

Step 1: Solve the linear approximation.
Step 2: Compute the actual objective value of FCNFP, Z_local.
Step 3: If $Z\_local < Z$,

i. $Z = Z\_local; x_{ij} = x\_local_{ij};$

ii. If $x\_local_{ij} > 0$ and $x\_pre_{ij} = 0$, include (i, j) into the inspiration set ζ;
iii. Update the linear factors of the other arcs, $\bar{c}^b_{ij} = \bar{c}_{ij}$.

Step 4: Record the solution information:

i. If $x\_local_{ij} > 0, t\_xnonzero\_sh_{ij} = t\_xnonzero\_sh_{ij} + 1$; $t\_xnonezero\_lm_{ij} = t\_xnonzero\_lm_{ij} + 1$;
ii. Compute the average reduced costs, $avg\_rc_{ij.}$

Step 5: If the process reaches the maximum iterations of DSSP phase or there are the same solutions in two consecutive iterations, go to Phase 2; otherwise, update the linear factor in the objective function of the linear approximation according to (10). Go to Step 2.

Phase 2: Intensification

Step 1: Set $t\_xnonzero\_sh_{ij} = 0$;
Step 2: Examine the tabu status of the arcs with $t\_xnonzero\_sh_{ij} \neq 0$:

i. If $tb_{ij} = tb\_max\_sh$, remove arc (i, j) from tabu set τ;
ii. If $tm_{ij} = tm\_max\_sh$, add arc (i, j) into τ.

Step 3: Move the arcs in ζ into the intensification candidate set α;
Step 4: Examine the other arcs with $t\_xnonzero\_sh_{ij} \geq k/2$;
If arc $(i, j) \notin \tau$. add arc (i, j) into α; $tm_{ij} = tm_{ij}+1$; otherwise, $tb_{ij} = tb_{ij}+1$.
Step 5: If the process reaches the maximum iterations of the intensification function, go to Phase 3; otherwise, update the linear factor according to (11) and go to Step1 of Phase 1.

Phase 3: Diversification

Step 1: Examine the tabu status of the arcs with $t\_xnonzero\_lm_{ij} \neq 0$:

i. If $tb_{ij} = tb\_max\_lm$, remove arc (i, j) from the tabu set τ;
ii. If $tm_{ij} = tm\_max\_lm$, add arc (i, j) into τ.

Step 2: Examine the arcs with $t\_xnonzero\_lm_{ij} = 0$:

i. Find the arcs with the 1st through the (*divno*)-th least average reduced cost, avg_rc_{ij} among the set of $\{(i, j)|(i, j) \notin \tau\}$;
ii. Move these arcs into the diversification set β. Set $tm_{ij} = tm_{ij} + 1$;
iii. For those arcs with the least avg_rc_{ij}, that belong to the tabu set τ, set $tb\_ij = tb_{ij} + 1$.

Step 3: If the process completes the maximum iterations of diversification, stop; otherwise update the linear factor according to (15) and go to step 1 of Phase 1.

5. Computational Experiment

The computational experiment was designed to test the performance of the enhanced DSSP in solving FCNFP problems, compared to the pure DSSP and CPLEX B&B. The enhanced DSSP, the pure DSSP and B&B method were implemented in C with callable library of CPLEX 7.0. The experiment was performed on a SunBlade UNIX machine with two 750 MHz 64-bit UltraSPARC-II processors and 1 GB of memory. The FCNFP problems are generated randomly in line with the convention of the difficult instances from the online OR test problem library.

5.1. TEST PROBLEMS

The test problems range from 50 nodes to 250 nodes with the density ranging from 0.1 to 0.5, i.e., the number of arcs ranging from 245 through 10973. The problems were divided into three groups in terms of the network density and, in each group, were further categorized into the types of problems with the different number of nodes. For each problem type, 6 to 12 test problems were randomly generated and solved. The design of the test problems is given in Table I. The column "Node" indicates the number of nodes in each test problem type. Let n denote the number of nodes and d denote the density of the test problem. The number of arcs in the column "Arc" is given as the integer part of $n \times (n - 1) \times d$. The column "Tested problems" shows the number of tested problems for each problem type. "Total in group" indicates the total number of tested problems for each group.

Table I. Test problem category.

Group	Node	Density	Arc	Tested problems	Total in group
1	50	0.10	245	12	45
	100		990	12	
	150		2235	9	
	180		3222	6	
	210		4398	6	
2	50	0.25	613	12	39
	100		2475	9	
	150		5588	6	
	180		8055	6	
	210		10973*	6	
3	50	0.50	1225	9	15
	100		4950	6	

*CPLEX B&B solved 6 problems out of 7 test problems. For the problem, B&B failed to obtain the exact solution, it reported out of memory after running 191455.24s (53.18 hours).

5.2. PARAMETERS

The enhanced DSSP uses the following six parameters:

- the size of MOVE set α for intensification;
- the number of times of an arc being a *tabu* in intensification;
- the number of times of an arc being a *move* in intensification;
- the size of MOVE set β for diversification;
- the number of times of an arc being a *tabu* in diversification;
- the number of times of an arc being a *move* in diversification.

We conducted a preliminary experiment to obtain some insight into the pure DSSP process in terms of active and non-active arcs. 5 to 10 problems were tested for each problem type to find the average active and non-active arcs in the pure DSSP. The preliminary experiment studies the solutions of the pure DSSP. It is found that the problems with density of 0.1–0.15 have on average 80–97% of arcs never visited in the pure DSSP, the problems with 0.15–0.25, 76–94%, and the problems with 0.25–0.5, 65–84%. In addition, it is observed that there is a relatively constant number of visited arcs regardless of individual problem instance in each type of problems, if we use the condition $t\_nonzerox\_sh_{ij} > k/2$. These arcs can be considered as the virtually frequently visited arcs in the initial DSSP and will be selected into α. Let r be the number of arcs with $t\_nonzerox\_sh_{ij} \geq k/2$ in the DSSP phase immediately before the current intensification process, which do not belong to the tabu set. Let s be the number of arcs in the aspiration set ζ. Consequently, the size of α, denoted as $intno = r + s$.

Two sets of parameters were tested with tb_max_sh:$tm\_max\_sh = 1 : 2$ and 2:1 in order to have the search process focus on the frequent and promising solution neighborhood.

The purpose of the diversification is to explore more various regions of the solution space that have never been visited by DSSP phases in the search history. The setting of *divno* is related to the number *m* of inactive arcs in the first DSSP phase of the enhanced DSSP. Note that the later DSSP phases are distorted by the penalties and incentives purposely imposed during the intensification and diversification. The process counts *m* during the first DSSP phase. It is obvious that the size of MOVE set β, *divno* should be increased as the problem size increases, so *divno* is set in terms of percentage of *m*. The settings of $m = 2\%$, 5%, 10% and 20% are tested.

Since there are considerably more non-basic arcs than basic arcs in the problems, we set tb_max_lm greater than tm_max_lm in the diversification process with tm_max_lm fixed to

1. Three settings were tested initially, including (*m*/*divno*): 1, (*m* /(*2divno*)): 1 and 1:1. Note that tb_max_lm is greater than the preset diversification iteration, (i.e. 20), when $divno < 5\% \times m$, which means some of the arcs will not be explored by any means at the end of the search process even if $tb\_max\_lm > 20$. Therefore, the maximum value of (*m*/*divno*) is set to 20. The results show that the latter two

Table II. Final experiment scenarios (S1–S8).

intno	$tb\_max\_sh : tm\_max\_sh$	*divno*	$tb\_max\_lm : tm\_max\_lm$	Scenario
$r + s$	1:2	0.02 m	20:1	S1
		0.05 m	20:1	S2
		0.1 m	10:1	S3
		0.2 m	5:1	S4
	2:1	0.02 m	20:1	S5
		0.05 m	20:1	S6
		0.1 m	10:1	S7
		0.2 m	5:1	S8

settings found inferior solutions during the entire experiment. Hence, we eliminate those settings to simplify the experimental scenarios. The final scenarios are given in Table II.

5.3. SOLUTION QUALITY

We adopted the relative errors to evaluate and compare the performances of the enhanced DSSP and the pure DSSP by comparing the solutions found by the two heuristics with the exact solutions found by B&B. The relative errors are evaluated as follows:

$$ERR_{\text{DSSP}} = \frac{f_{\text{DSSP}} - f_{\text{exact}}}{f_{\text{exact}}}$$

$$ERR_{\text{DSSP\_tabu}} = \frac{f_{\text{DSSP\_tabu}} - f_{\text{exact}}}{f_{\text{exact}}}.$$

The numerical results are shown in Table III–V (group 1–3) separated in the eight scenarios of the enhanced DSSP. The tables show the performance comparison between the pure DSSP and the enhanced DSSP in terms of the average, minimum and maximum relative errors compared with the exact solutions and the number of exact solutions found by the pure DSSP and the enhanced DSSP. The column "Number" represents the number of test problems. The columns labeled "Exact" indicate the number of problems for which the exact solution was found by the pure DSSP or the enhanced DSSP.

Tables III, IV,and V indicate that in the sparse networks, both the enhanced DSSP and the pure DSSP found good solutions. As the network size and, particularly, the density increase, it becomes harder for both heuristics to find the exact solutions, and the solutions obtained by the two heuristics deteriorate slightly. Nonetheless, except in the problems with 210×0.25, the mean relative errors of the solutions from the enhanced DSSP are still fairly stable as the problems become more difficult,

Table III. Relative error comparison between the pure DSSP and the enhanced DSSP [Group 1].

Problem		DSSP				DSSP_Tabu				
Type (# of arcs)	Number	Mean	Min	Max	Exact	Scenario	Mean	Min	Max	Exact
50 × 0.1 (245)	12	0.0005	0.0000	0.0063	11	S1	0.0000	0.0000	0.0000	12
						S2	0.0000	0.0000	0.0000	12
						S3	0.0000	0.0000	0.0000	12
						S4	0.0000	0.0000	0.0000	12
						S5	0.0000	0.0000	0.0000	12
						S6	0.0000	0.0000	0.0000	12
						S7	0.0000	0.0000	0.0000	12
						S8	0.0000	0.0000	0.0000	12
100 × 0.1 (990)	12	0.0163	0.0000	0.0550	4	S1	0.0148	0.0000	0.0538	6
						S2	0.0148	0.0000	0.0538	6
						S3	0.0148	0.0000	0.0538	6
						S4	0.0148	0.0000	0.0538	6
						S5	0.0148	0.0000	0.0538	6
						S6	0.0148	0.0000	0.0538	6
						S7	0.0148	0.0000	0.0538	6
						S8	0.0148	0.0000	0.0538	6
150 × 0.1 (2235)	9	0.0125	0.0000	0.0650	4	S1	0.0122	0.0000	0.0643	5
						S2	0.0104	0.0000	0.0480	5
						S3	0.0096	0.0000	0.0643	6
						S4	0.0055	0.0000	0.0228	5
						S5	0.0117	0.0000	0.0650	5
						S6	0.0106	0.0000	0.0498	5
						S7	0.0055	0.0000	0.0228	5
						S8	0.0104	0.0000	0.0480	5
180 × 0.1 (3222)	6	0.0155	0.0000	0.0375	1	S1	0.0130	0.0000	0.0315	2
						S2	0.0130	0.0000	0.0315	2
						S3	0.0130	0.0000	0.0315	2
						S4	0.0130	0.0000	0.0315	2
						S5	0.0101	0.0000	0.0315	2
						S6	0.0095	0.0000	0.0315	2
						S7	0.0095	0.0000	0.0315	2
						S8	0.0102	0.0000	0.0315	2
210 × 0.1 (4398)	6	0.0175	0.0077	0.0285	0	S1	0.0142	0.0000	0.0254	1
						S2	0.0142	0.0000	0.0254	1
						S3	0.0142	0.0000	0.0254	1

(*Continued on next page*)

Table III. (Continued)

Problem		DSSP				DSSP_Tabu				
Type (# of arcs)	Number	Mean	Min	Max	Exact	Scenario	Mean	Min	Max	Exact
						S4	0.0142	0.0000	0.0254	1
						S5	0.0133	0.0000	0.0254	1
						S6	0.0142	0.0000	0.0254	1
						S7	0.0153	0.0067	0.0254	0
						S8	0.0142	0.0000	0.0254	1

changing from 0.00% to 1.48% in the best scenarios and 0.00% to 1.82% in the worst scenarios; while the relative errors for the pure DSSP increase somewhat dramatically, from 0.05% to 3.50%. Furthermore, based on the average relative errors, the improvements in the solution quality of the enhanced DSSP over the pure DSSP in the best scenarios (S1–S8) and the worst scenarios for each type of problems are shown in Table VI. Two columns with heading "*Si*" represent the scenarios in the enhanced DSSP that the least/most relative errors appear. The columns "Qual. Imp." represent the percentage of the solution quality improvement by comparing the mean relative errors of the enhanced DSSP, denoted as $\overline{ERR}_{\text{dssp\_tabu}}$ and the mean relative errors of the pure DSSP for each problem type, denoted as $\overline{ERR}_{\text{dssp}}$. In other words, the "Qual. Imp." is computed as:

$$\frac{\overline{ERR}_{\text{dssp}} - \overline{ERR}_{\text{dssp\_tabu}}}{\overline{ERR}_{\text{dssp}}}.$$

The improvement in solution quality of the enhanced DSSP over the pure DSSP is over 20% in the best scenarios and over 10% in the worst scenarios. This shows that the enhanced DSSP provides substantial improvement for most of tested problems.

5.4. CPU TIMES

Let $T_{\text{b\&b}}$, T_{dssp}, and $T_{\text{dssp\_tabu}}$ be the CPU times in seconds used by the CPLEX B&B, the pure DSSP and the enhanced DSSP, respectively. The comparison of CPU times is given in Table VII. We averaged CPU times of all 8 scenarios in the enhanced DSSP as there is no significant difference of solution time among them.

CPLEX is well known for its fast MIP optimizer with multiple cutting-edge techniques involved in B&B method. However, the CPU times used by CPLEX B&B increase exponentially as the problem size and the network density increase. For the problems with similar sizes (e.g. 150×0.1 with 2235 arcs and 100×0.25 with 2475 arcs), the ones with higher density are more difficult for B&B to solve.

Table IV. Relative error comparison between the pure DSSP and the Enhanced DSSP [Group2].

Problem		DSSP				DSSP_Tabu				
Type (# of arcs)	Number	Mean	Min	Max	Exact	Scenario	Mean	Min	Max	Exact
50 × 0.25 (613)	12	0.0112	0.0000	0.0643	5	S3	0.0040	0.0000	0.0280	8
						S4	0.0040	0.0000	0.0280	8
						S5	0.0040	0.0000	0.0280	8
						S6	0.0040	0.0000	0.0280	8
						S7	0.0041	0.0000	0.0280	8
						S8	0.0041	0.0000	0.0280	8
100 × 0.25 (2475)	9	0.0350	0.0000	0.1312	1	S3	0.0170	0.0000	0.0435	1
						S4	0.0169	0.0000	0.0435	1
						S5	0.0150	0.0000	0.0416	2
						S6	0.0117	0.0000	0.0344	2
						S7	0.0149	0.0000	0.0357	1
						S8	0.0130	0.0000	0.0344	1
150 × 0.25 (5588)	6	0.0236	0.0032	0.0335	0	S1	0.0141	0.0000	0.0332	1
						S2	0.0155	0.0000	0.0332	1
						S3	0.0138	0.0000	0.0332	1
						S4	0.0136	0.0000	0.0332	1
						S5	0.0154	0.0000	0.0313	1
						S6	0.0137	0.0000	0.0332	1
						S7	0.0127	0.0000	0.0332	1
						S8	0.0144	0.0000	0.0332	1
180 × 0.25 (8055)	6	0.0177	0.0047	0.0377	0	S1	0.0132	0.0031	0.0248	0
						S2	0.0131	0.0031	0.0248	0
						S3	0.0123	0.0006	0.0272	0
						S4	0.0117	0.0031	0.0248	0
						S5	0.0144	0.0031	0.0269	0
						S6	0.0145	0.0031	0.0281	0
210 × 0.25 (10973)	6	0.0332	0.0080	0.0463	0	S1	0.0199	0.0057	0.0353	0
						S2	0.0227	0.0080	0.0424	0
						S3	0.0194	0.0080	0.0340	0
						S4	0.0236	0.0080	0.0389	0
						S5	0.0245	0.0080	0.0410	0
						S6	0.0296	0.0080	0.0424	0

When the number of arcs in a problem is close to/exceeds 5000, the problems with high density of 0.25 and 0.5 become particularly hard for B&B, and it becomes impractical to apply B&B. In addition, there is a large variation in the CPU time that is required by B&B to find the exact solutions. One extreme case is that B&B

Table V. Relative error comparison between the pure DSSP and the Enhanced DSSP [Group3].

Problem		DSSP				DSSP_Tabu				
Type (# of arcs)	Number	Mean	Min	Max	Exact	Scenario	Mean	Min	Max	Exact
50 × 0.5 (1225)	9	0.0228	0.0088	0.0403	0	S1	0.0171	0.0021	0.0351	0
						S2	0.0167	0.0021	0.0351	0
						S3	0.0159	0.0021	0.0311	0
						S4	0.0167	0.0021	0.0350	0
						S5	0.0169	0.0064	0.0281	0
						S6	0.0173	0.0021	0.0351	0
						S7	0.0132	0.0021	0.0350	0
						S8	0.0182	0.0064	0.0351	0
100 × 0.5 (4950)	6	0.0218	0.0000	0.0416	1	S1	0.0195	0.0000	0.0391	1
						S2	0.0208	0.0000	0.0370	1
						S3	0.0186	0.0000	0.0365	1
						S4	0.0152	0.0000	0.0365	1
						S5	0.0176	0.0000	0.0365	1
						S6	0.0166	0.0000	0.0365	1

Table VI. Solution quality improvement of the enhanced DSSP over the pure DSSP.

		Best scenario		Worst scenario	
Group	Type	*Si*	Qual. Imp.(%)	*Si*	Qual. Imp.(%)
1	50 × 0.1	S1–S8	100	S1–S8	100
	100 × 0.1	S1–S8	9.0	S1–S8	9.0
	150 × 0.1	S4–S7	55.9	S1	2.3
	180 × 0.1	S6–S7	38.7	S1–S4	16.1
	210 × 0.1	S5	24.4	S7	12.9
2	50 × 0.25	S3–S6	64.7	S7–S8	63.5
	100 × 0.25	S6	66.7	S3	51.5
	150 × 0.25	S7	46.1	S2	34.6
	180 × 0.25	S4	34.1	S6	18.0
	210 × 0.25	S3	41.5	S6	11.0
3	50 × 0.5	S7	42.0	S8	20.1
	100 × 0.5	S3	30.2	S2	4.4

used 113.68 hours (about 5 days) to solve a test problem in one of 210 × 0.25 problem category. By contrast, the pure DSSP and the enhanced DSSP used much less CPU time. The enhanced DSSP used more CPU time than the pure DSSP. However, since the increase in CPU time is of a relatively small magnitude (only 10–20 minutes more than the pure DSSP, yet hours or even days less than B&B),

Table VII. CPU times.

			$T_{b\&b}(S)$			$T_{dssp}(S)$			$T_{dssp\_tabu}(S)$		
Group	Problem	Arc	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min
1	50 × 0.1	245	0.39	3.88	0.04	0.02	0.12	0.00	0.48	3.66	0.03
	100 × 0.1	990	1.82	6.39	0.33	0.09	0.37	0.02	1.70	3.20	1.05
	150 × 0.1	2235	11.57	24.89	5.58	0.33	0.65	0.10	8.69	14.29	5.71
	180 × 0.1	3222	64.70	257.99	10.81	3.51	9.48	0.19	57.68	177.82	12.63
	210 × 0.1	4398	1955.08	7124.66	467.87	18.62	36.58	7.51	238.52	488.99	49.39
2	50 × 0.25	613	0.63	1.93	0.15	0.04	0.08	0.01	0.91	1.34	0.21
	100 × 0.25	2475	49.33	176.04	11.53	3.68	9.52	0.95	36.55	83.88	13.09
	150 × 0.25	5588	6087.03	18366.12	25.47	33.99	43.18	21.55	424.70	641.76	230.08
	180 × 0.25	8055	9765.63	35910.00	1769.50	91.41	155.66	19.22	938.91	1803.87	286.65
	210 × 0.25	10973	119612.95	409230.42	581.00	185.00	318.09	79.19	899.37	1529.63	356.62
3	50 × 0.5	1225	222.48	1070.97	8.05	1.20	2.11	0.60	14.04	33.52	4.06
	100 × 0.5	4950	1286.08	4113.58	268.25	22.01	43.24	2.43	213.64	430.18	76.70

and taking into account the solution quality improvement, we can conclude that the enhanced DSSP has its advantage.

6. Conclusions

In the paper, an enhanced DSSP algorithm with the tabu search scheme is presented for solving FCNFP problems. The algorithm integrates the pure DSSP and the tabu mechanism including intensification and diversification with short-term and long-term memory, respectively. The short-term memory based on recency, quality and frequency is used for the intensification process in selecting trigger arcs, while the concept of reduced cost and long-term memory based on recency and infrequency are used for the diversification process in selecting trigger arcs. The linear factors of these arcs are the imposed incentives and hence lead DSSP search process deeper or divert the search for better solutions. The results of computational experiments performed on large and various sets of problem instances show the efficiency of the enhanced DSSP compared with the pure DSSP and B&B in terms of the solution quality and CUP times.

In real world applications, B&B approach can be used if the problem is not too complex (small number of nodes and low density). However, if the problem becomes large scale and has high density, heuristic methods would be a better option for finding the solution. The computational experiments have demonstrated that the proposed enhanced DSSP is competitive in solving larger and difficult problems. For many cases, we recommend to use a combination of two or more different scenarios (S1–S12) to attack the same problem, since none of single scenarios could guarantee finding better solutions all the time. The best solution obtained by the proposed heuristic can serve as the upper bound of the exact algorithm in finding the optimal solution, which will drastically decrease the search time of the exact solution.

References

Barr, R.S., Glover, F. and Klingman, D. (1981). A new optimization method for large scale fixed charge transportation problems. *Operation Research*, **29**, 448–463.

Cabot, A.V. and Erenguc, S.S. (1984). Some branch-and-bound procedures for fixed-cost transportation problems. *Naval Research Logistics Quarterly*, **31**, 145–154.

Crainic, T.G., Frangioni, A. and Gendron, B. (2001). Bundle-based relaxation methods for multicommodity capacitated fixed charge network design. *Discrete Applied Mathematics*, **112**, 73–99.

Crainic, T.G., Gendreau, M. and Farvolden, J. (2000). A Simplex-based tabu search Method for capacitated network design. INFORMS, *J. Computing*, **12**(3), 223-236.

Gendron, B. and Crainic T.G. (1994). Relaxation for multicommodity capacitated network design problems Publication CRT-965, Centre de Recherche Sur Les Transports, University of Montreal,

Geoffrion, A.M. (1974). Lagrangean relaxation for integer programming. *Mathematical Programming Study*, **2**, 82–114.

Glover, F. (1994). Optimization by ghost image processes in neural networks. *Computers & Operations Research*, **21**(8), 801–822.

Glover, F. (1989). Tabu search: Part I ORSA. *Journal on Computing*, **1**(3), 190–206.

Glover, F. (1990). Tabu search: Part II ORSA. *Journal on Computing*, **2**(1), 4–32.

Glover, F. and Laguna, M. (1997). *Tabu Search*, Kluwer Academic Publishers, Hingham, MA,

Gray, P. (1971). Exact solution of the fixed-charge transportation problem. *Operations Research*, **19**(6), 1529–1538.

Hochbaum D.S. and Segev, A. (1989). Analysis of a flow problem with fixed charges. *Networks*, **19**, 291–312.

Kim, D. and Pardalos, P.M. (1999,). A solution approach to the fixed charge network flow problem using a dynamic slope scaling procedure. *Operations Research Letters*, **24**, 195–203.

Magnanti, T. L., Mireault, P. and Wong, R.T. (1986). Tailoring Benders decomposition for uncapacitated network design. *Mathematical Programming Study*, **26**, 112–154.

Palekar, U.S., Karwan, M.K. and Zionts, S. (1990). A branch-and-bound method for the fixed charge transportation problem. *Management Science*, **36**(9), 1092–1105.

Steinberg, D.I. (1970). The fixed charge problem. *Naval Research Logistics Quarterly*, **17**, 217–235

Suhl, U. (1985). Solving large scale mixed integer programs with fixed charge variables. *Mathematical Programming*, **32**, 165–182.

Sun, M., Aronson, J.E., McKeown, P.G. and Drinka, D. (1998). A tabu search heuristic procedure for the fixed charge transportation problem. *European Journal of Operational Research*, **106**, 441–456.