HYBRIDISING LOCAL SEARCH WITH BRANCH-AND-BOUND FOR CONSTRAINED PORTFOLIO SELECTION PROBLEMS

Fang He[1, 2] and Rong Qu[1]
1 The Automated Scheduling, Optimisation and Planning (ASAP) Group, School of Computer Science
The University of Nottingham, Nottingham, NG8 1BB, UK
2 Department of Computer Science, Faculty of Science and Technology, University of Westminster, W1B 2HW, UK
Email: hef@westminster.ac.uk, rxq@cs.nott.ac.uk

KEYWORDS
Hybrid algorithm; Branch-and-Bound; local search; portfolio selection problems

ABSTRACT

In this paper, we investigate a constrained portfolio selection problem with cardinality constraint, minimum size and position constraints, and non-convex transaction cost. A hybrid method named ***Local Search Branch-and-Bound*** (LS-B&B) which integrates local search with B&B is proposed based on the property of the problem, i.e. cardinality constraint. To eliminate the computational burden which is mainly due to the cardinality constraint, the corresponding set of binary variables is identified as core variables. *Variable fixing* (Bixby, Fenelon et al. 2000) is applied on the core variables, together with a local search, to generate a sequence of simplified sub-problems. The default B&B search then solves these restricted and simplified sub-problems optimally due to their reduced size comparing to the original one. Due to the inherent similar structures in the sub-problems, the solution information is reused to evoke the repairing heuristics and thus accelerate the solving procedure of the sub-problems in B&B. The tight upper bound identified at early stage of the search can discard more sub-problems to speed up the LS-B&B search to the optimal solution to the original problem. Our study is performed on a set of portfolio selection problems with non-convex transaction costs and a number of trading constraints based on the extended mean-variance model. Computational experiments demonstrate the effectiveness of the algorithm by using less computational time.

INTRODUCTION

In this paper, we tackle the single-period portfolio selection problem (PSP). In the problem concerned, a number of transactions can be carried out to adjust the portfolio during a given trading period. We take into account these transaction costs as well as a set of trading constraints. These include the cardinality constraint (a limit on the total number of assets held in the portfolio, i.e. select k out n ($k<n$) assets to be held in the portfolio), the minimum position size constraint (bounds on the amount of each asset), the minimum trade size constraint (bounds on the amount of transaction occurred on each asset) and transaction costs. The goal of the problem is to minimize the risk of the adjusted portfolio and the transaction costs incurred, while satisfying the set of trading constraints in feasible portfolios. The aim of this paper is to develop a hybrid method to solve the complex PSP efficiently. The techniques developed here are employed to solve a specific problem, but it could be applied to other variants of PSP with cardinality constraint, and possible other combinatorial problems outside this domain.

If the transaction cost function is linear, then the problem is generally easy to solve. However, a function which better reflects realistic transaction costs is usually non-convex (Konno and Wijayanayake 2001). Some research show that realistic transaction costs usually include a fixed fee, and thus the cost is relatively higher when the amount of transaction is smaller (Konno and Wijayanayake 2001, Konno and Wijayanayake 2002). The transaction cost is thus usually represented by a linear piecewise concave function. This turns the problem into a non-convex optimisation problem, which is more difficult to solve.

In this paper, we propose a new hybrid approach which integrates local search with B&B to solve the non-convex portfolio selection problem *heuristically*. We conceptually divide the decision variables into two parts: the set of core variables which defines the cardinality constraint and the rest of variables. Variable fixing is applied to the core variables. The result of variable fixing has two facets: values (i.e. 0, 1) are assigned to the core binary variables and simplified sub-problem is generated. A local search together with variable fixing are performed on the core variables to generate a sequence of simplified sub-problems. These sub-problems are traversed heuristically to find the promising sub-problems, i.e. whose lower bounds are not greater than upper bounds. The promising sub-problems then are solved by a default B&B. Value assignments by variable fixing, together with the value assignments by a default B&B, form the complete solutions to the original problem. The best solution to the sub-problem, together with the value assignments by variable fixing approximates an optimal solution to the original problem.

PROBLEM FORMULATION

Consider that an investor is holding an initial portfolio that consists of a set of n assets. To respond to the changes in the market, the investor must review its current portfolio, with the view to carry out a number of transactions. It is assumed that the new portfolio will be held for a fixed time period. The investor's goal is to minimize both the transaction costs occurred and the risk of the assets in the portfolio at the end of the investment period, while satisfying a set of constraints. These constraints typically include meeting the target return, the minimum position size, and the minimum trading size.

Let w_i be the percentage of capital invested in asset i, $i=1,...,n$. We shall use a weight vector $\boldsymbol{w^0} = (w_1^0, w_2^0, ..., w_n^0)^T$ to denote an initial portfolio. The percentage amount transacted in each asset is specified by weight vector $\boldsymbol{x} = (x_1, x_2..., x_n)^T$, $x_i < 0$ means selling and $x_i > 0$ means buying. A weight vector $\boldsymbol{w}$ denotes the portfolio after the revision. After the transaction, the adjusted portfolio is $\boldsymbol{w} = \boldsymbol{w^0} + \boldsymbol{x}$, and is held for a fixed period of time. We denote the return of asset i at the end of the investment period as r_i and the expected return of the portfolio as R. We denote the covariance between assets i and j in return as σ_{ij}. We further define $\phi(x)$ as the sum of individual transaction costs associated with each x_i. Based on the basic MV model, the portfolio selection problem with transaction costs can thus be modeled as follows:

$$\min \sum_{i=1}^{i=n}\sum_{j=1}^{j=n}\sigma_{ij}w_i w_j + \phi(x) \tag{1}$$

$$s.t. \quad \sum_{i=1}^{i=n} r_i w_i = R \tag{2}$$

$$\boldsymbol{w} = \boldsymbol{w^0} + \boldsymbol{x} \in F \tag{3}$$

where objective (1) is to minimize the risk of the portfolio and the transaction costs incurred. (2) ensures the expected return. F in (3) represents a set of feasible portfolios subject to all the related constraints. These constraints include the minimum position size, the minimum trading size, etc., which will be detailed next. In this paper, we model the problem as a single objective problem where (1) is the sum of two objects with the same weights.

The transaction cost is the sum of the transaction costs associated with the assets traded:

$$\phi(x) = \sum_{i=1}^{i=n}\phi_i(x_i)$$

In this paper, we consider a model that includes a fixed fee plus a linear cost, thus leads to a non-convex function, as shown in Fig. 1. This function is also applied in (Lobo, Fazel et al. 2007) . The fixed fee charged for buying and selling asset i is denoted as β_i^+ and β_i^- , and the variable costs associated to buying and selling asset i are denoted by α_i^+ and α_i^- . The transaction cost function is given in (4), and shown in Fig. 1:

$$\phi_i(x_i) = \begin{cases} 0, x_i = 0; \\ \beta_i^+ + \alpha_i^+ x_i , x_i > 0; \\ \beta_i^- - \alpha_i^- x_i , x_i < 0; \end{cases} \tag{4}$$

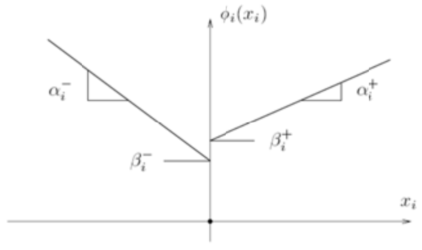

Fig.1 The transaction cost function (Lobo, Fazel et al. 2007)

Problem Model with Transaction Cost and Trading Constraints

Parameter

n	The total number of assets
i	The index of assets, $i=1,...,n$
w^0	Initial position of the portfolio
σ_{ij}	Covariance between assets i and j
r_i	Return of asset i at the end of the investment period
R	Expected return of the portfolio
β_i	Fixed cost for buying or selling asset i
α_i	Variable cost rate for buying or selling asset i

$w_{\min}$	Minimum hold position
$x_{\min}$	Minimum trading amount
k	Number of assets in the portfolio after transaction

Variable		Feature
w_i	Revised position of the portfolio after transaction	Decision variable
x_i^{buy}	Amount of buying asset *i*	Decision variable
x_i^{sell}	Amount of selling asset *i*	Decision variable
z_i	Hold asset *i* or not in the revised portfolio	Auxiliary variable
z_i^{buy}	Buy asset *i* or not	Auxiliary variable
z_i^{sell}	Sell asset *i* or not	Auxiliary variable

$$\min \sum_{i=1}^{i=n}\sum_{j=1}^{j=n}\sigma_{ij}w_iw_j + \sum_{i=1}^{i=n}\phi_i(x_i) \qquad (1) \qquad \text{(PSP)}$$

$$s.t.$$

$$\sum_{i=1}^{i=n} r_iw_i = R \qquad (2)$$

$$w_i = w_i^0 + x_i^{buy} - x_i^{sell}, i=1,...n \qquad (3)$$

$$\sum_{i=1}^{i=n} w_i + \sum_{i=1}^{i=n}\phi_i(x_i) = 1 \qquad (5)$$

$$w_i \le z_i \ , i=1,...n \qquad (6)$$

$$w_{\min} z_i \le w_i \ , i=1,...n \qquad (7)$$

$$\sum_{i=1}^{i=n} z_i = k \qquad (8)$$

$$x_{\min} z_i^{buy} \le x_i^{buy}, i=1,...n \qquad (9)$$

$$x_{\min} z_i^{sell} \le x_i^{sell}, i=1,...n \qquad (10)$$

$$x_i^{sell} \le z_i^{sell}, i=1,...n \qquad (11)$$

$$x_i^{buy} \le z_i^{buy}, i=1,...n \qquad (12)$$

$$z_i^{buy} \le z_i \ , i=1,...n \qquad (13)$$

$$z_i^{buy} + z_i^{sell} \le 1 \ , i=1,...n \qquad (14)$$

$$0 \le w_i \le 1, i=1,...n \qquad (15)$$

$$0 \le x_i^{buy} \le 1, i=1,...n \qquad (16)$$

$$0 \le x_i^{sell} \le 1, i=1,...n \qquad (17)$$

$$z_i, z_i^{buy}, z_i^{sell} \in \{0,1\}, i=1,...n \qquad (18)$$

There are two groups of variables in the formulation of the problem, as denoted by the "feature" column. w_i, x_i^{buy} , x_i^{sell} are decision variables. z_i , z_i^{buy} and z_i^{sell} are auxiliary variables which are used to formulate the constraints. The column "core variable" denotes which variables are core variables. The selection of the core variables is problem dependent. Several researchers have pointed out that the cardinality constraint presents the greatest computational challenge to the problem (Bienstock 1996, Jobst, Horniman et al. 2001, Stoyan and Kwon 2010, Stoyan and Kwon 2011). Actually, the PSP with cardinality constraint has been recognized to be NP-complete (Bienstock 1996, Mansini and Speranza 1999). To eliminate the cardinality constraint, we identify variables z_i which define the cardinality constraint $\sum_{i=1}^{i=n} z_i = k$ as a set of core variables.

Based on the model PSP, we will introduce two additional reduced models (PSP basic, PSP sub) as follows which will be applied to evaluate the neighbourhood in the local search and to calculate the lower bound:

$$\min \sum_{i=1}^{i=n}\sum_{j=1}^{j=n}\sigma_{ij}w_iw_j \qquad (1) \qquad \text{(PSP basic)}$$

$$s.t.$$

$$\sum_{i=1}^{i=n} r_iw_i = R \qquad (2)$$

$$w_i \le z_i \ , i=1,...n \qquad (6)$$

$$w_{\min} z_i \le w_i \ , i=1,...n \qquad (7)$$

$$\sum_{i=1}^{i=n} z_i = k \qquad (8)$$

$$0 \le w_i \le 1, i=1,...n \qquad (15)$$

$$z_i \textit{ with assignments in } \{0,1\}, i=1,...n \quad (18)$$

$$\min \sum_{i=1}^{i=n}\sum_{j=1}^{j=n}\sigma_{ij}w_iw_j + \sum_{i=1}^{i=n}\phi_i(x_i) \qquad (1) \qquad \text{(PSP sub)}$$

$$s.t.$$

$$(2)-(17)$$

$$z_i \textit{ with assignments in } \{0,1\}, i=1,...n \qquad (18)$$

$$z_i^{buy}, z_i^{sell} \in \{0,1\}, i=1,...n \qquad (19)$$

LS-B&B TO PSP ALGORITHM

In this section, we propose a new hybrid search, named LS-B&B to PSP according to the property of the problem. To the PSP with binary variable z_i we are dealing with, we know that exactly k out n binary variables will be assigned to 1 in the feasible and optimal solutions. With this knowledge, we can apply variable fixing on a set of variables at one time, resulting into simplified sub-problem. A local search is performed on these set of variables to generate a sequence of sub-problems, and the best solution will be identified among them.

Framework of LS-B&B to PSP

We present the framework of LS- B&B to PSP, as shown in Fig.2.

LS-B&B consists of four main components. The first component is the initialization phase (line 1). In this phase, variable fixing is applied to the core variables to generate a simplified sub-problem. Lower bound and upper bound of the problem are also initialized in this phase.

The second component is a default B&B search (line 7). It is called to solve the sub-problems to optimality. This solution to the sub-problem together with the variable assignments by variable fixing, forms the solution to the original problem.

The third component is a local search (line 9) which is performed on set Z of variable z_i to update sets S and. With the updated S, the sub-problem is updated correspondingly. Therefore, we state that this local search generates a sequence of sub-problems.

The fourth component is an overall search procedure (the while loop). In this search procedure, a local search, variable fixing and a default B&B work together to identify the best solution among the sub-problems by pruning inferior sub-problems and solving the promising sub-problems to optimality.

We present explanations of these components next.

Components of LS-B&B to PSP

Variable fixing

(Hard) variable fixing has been used in MIP context to divide a problem into sub-problems. It assigns values to a subset of variables of the original problem. That is, certain variables are fixed to the given values. Based on the definition of variable fixing in (Bixby, Fenelon et al. 2000, Lazic, Hanafi et al. 2009), we apply this variable fixing to simplify the original problem into sub-problems in the following way. We first denote a subsets S on the binary variable set B: $S \subseteq B$. Then we fix variables in subsets S to 1, to obtain sub-problems P_{sub_y} as follows:

$$\begin{aligned} &P_{sub_y} : \min c^T x \\ &s.t. Ax \le b; \\ &x_j = 1, \forall j \in S \subseteq B \ne \varnothing \\ &x_j = [0,1], \forall j \in C \end{aligned}$$

In this way, we simplify the original problem to a sub-problem. One selection of the subsets S can generate one possible simplified sub-problem of the original problem. Therefore, we apply variable fixing together with a local search to generate a sequence of sub-problems where we will search for the best solution.

```
LS- B&B
LB: lower bound;
UB: upper bound;
(h, x, w, z): a solution (x, w, z) of the problem with a
        corresponding objective value h;
solveB&B: a default B&B solver;
Z: set of $z_i$;
S: subset of Z;
$P_{org}$: the original problem defined by model (PSP);
$P_{sub_y}$ : sub-problem defined by variable fixing;

1: Initialization phase
2: while (the number of iterations not met)
3:      If ($LB(P_{sub_y}) \ge UB$)
4:           prune the sub-problem $P_{sub_y}$ ;
5:           go to line 9;
6:      Else
7:           $(h, x, w, z)$ = solveB&B($P_{sub_y}$) ;
8:           if $h < UB$ set $UB = h$;
9:      perform a Local search on set Z; 10:
        generate sub-problems by variable fixing: $P_{sub_y} = P_{org} \cup (z_i = 1), z_i \in S$;
11: set $(x^*, w^*, z^*)$ as the best solution among all $(x, w, z)$
        and $h^*$ be the corresponding objective value;
```

Fig. 2 The LS-B&B algorithm to PSP

Initialization phase

The main task of the initialization phase is the generation of a sub-problems P_{sub_y} by variable fixing on variables z_i on sets S. From the definition of P_{sub_y}, we can state that P_{sub_y} is P_{org} with the initialization of variables in S to 1.

In the initialization phase, the lower bound is obtained by solving the continuous relaxation of the sub-problem P_{sub_y} based on model (PSP sub), and the upper bound is set as ∞.

Default B&B search

As we stated in the framework of LS-B&B, each of the sub-problems itself is still a MIQP problem due to the presence of binary variables z_i^{buy} and z_i^{sell}. However, due to the assignments of variable z_i by variable fixing, the size of the sub-problem is much smaller comparing to the original one. Therefore, sub-problems can be handled by the default B&B. In this paper, the default B&B algorithm in the MIQP solver in CPLEX is applied to solve the promising sub-problems (when LB (P_{sub_y}) < UB) to optimality. What is more, the inherent similar structures of the sub-problems enable a very successful reuse of solution information, so the repairing heuristics embedded in solveB&B are evoked to improve the search.

Overall search procedure

The overall search explores the sequence of sub-problems. This is shown in the while loop in Fig.2. In this search, the lower bound of the sub-problem P_{sub_y} is computed by a general QP solver, which relaxes the sub-problem to a continuous problem, i.e. model PSP sub (line 3 in Fig.2). Here, the computation of the lower bound is different from the evaluation of a solution in the local search, which is based on model PSP basic. The objective value of the feasible solution to the concerned sub-problem P_{sub_y} serves as the upper bound of the original problem. If the lower bound of a sub-problem is above the current upper bound found so far, we can discard this sub-problem during the search (line 4 in Fig.2). Otherwise, these promising sub-problems are solved exactly by a default B&B (line 7 in Fig.2). The solutions to the sub-problems together with the assignments of core variables consist of the feasible solutions to the complete original problem. These sub-problems are solved in sequence, and the best solution among them, together with the variable assignments done by variable fixing, approximates the optimal solution to the original problem. The whole procedure terminates by a pre-defined number of iterations in the local search. Therefore, the search is an incomplete search. It cannot guarantee optimality of the solution due to the nature of the local search on core variables z_i.

The local search together with variable fixing creates a sequence of sub-problems which have very similar structures. They only differ in the coefficient or the right-hand side of constraints which are related to z_i. When solving this sequence of sub-problems, the solution information such as the basis list and basis factors from its simplex tableau (i.e., we apply the extended tableau simplex algorithm in the default MIQP solver) for the current problem are stored, and this can be retrieved and applied to the successive sub-problems. This means the solution information (i.e., basis list and basis factors) of the problem P_{sub_y} can thus be reused to obtain solution to $P_{sub_{y'}}$, so that $P_{sub_{y'}}$ does not need to be solved again from scratch. This solution information reusing thus can evoke the repairing heuristics embedded in the default B&B solver. This solution information reusing has shown to be extremely efficient.

EXPERIMENTAL RESULTS

To evealuate our algorithm on more general benchmark instances, we also concern in this paper the portfolio optimisation instances publicly available in the OR library (ORlibrary), with additional constraints derived from the above real-world problem. Six problem instances are used to test the algorithm proposed in this paper, which can be found at (He and Qu, 2014).

We set the minimum proportion of wealth to be invested in an asset, w_{min}, to 0.01%, and the minimum transaction amount, x_{min}, to 0.01%. We also set the parameters in the transaction cost function α_i to 0.005 and β_i to 0.0001 for all the assets. Other values of k in the cardinality constraint have been tested, ranging from 10 to 150 for different sizes of portfolios.

Evaluations on the LS-B&B algorithm

In LS-B&B, after fixing values for variables z_i by variable fixing and the local search, the resulting MIQP sub-problems are created. If the lower bound of a sub-problem is not greater than the current upper bound (we say it is a promising sub-problem, otherwise it will be pruned), it will be solved by the default B&B in CPLEX12.0. Therefore, when these sub-problems are processed, in conclusion four possible situations could emerge: (1) a sub-problem could be solved by B&B to optimality; (2) the repairing heuristic mechanism imbedded in CPLEX could be evoked and applied to a sub-problem to obtain a feasible solution heuristically; (3) a sub-problem could be pruned; this will happen if the optimal solution under continuous relaxation on model PSP sub is larger than the current upper bound; and (4) the solution of a sub-problem could be infeasible.

Table 1 illustrates the behavior of the above four situations during the processing of sub-problems. The total CPU time of the algorithm is dependent upon the CPU time needed for each situation.

Table 1. Information of sub-problem processing.

Instance	total CPU time	sub-problem solved		sub-problem repaired		sub-problem pruned		sub-problem infeasible	
		Number	Avg CPU time/p	Numb	Avg CPU time/p	Number	Avg CPU time/p	Numb	Avg CPU time/p
Société Générale	3.16	56	0.01	398	0.006	86	0	60	0
HangS	3.09	184	0.01	178	0.005	120	0	118	0
DAX	9.00	296	0.02	121	0.01	112	0.01	71	0
FTSE	11.44	79	0.08	102	0.025	127	0.02	292	0
S&P	13.55	286	0.04	114	0.01	77	0	123	0
Nikkei	76.97	89	0.40	21	0.36	221	0.08	269	0.06

Table 1 clearly indicates that the CPU time for identifying infeasibility is negligible. The CPU time for pruning the inferior sub-problem is quite efficient. Therefore, the more sub-problems pruned, the more efficient the search is. It can be interpreted from Table 1 that solving sub-problems with repairing heuristics is quite efficient. These repairing heuristics are the results of solution information reuse in the B&B solver. Solving sub-problems exactly is the most time consuming situation comparing with the other three situations.

Comparisons with the default B&B in CPLEX

It is worth noting that LS-B&B is a heuristic approach to the problem. It cannot prove optimality of the solution due to the nature of the local search on core variables z_i, although the sub-problems can be measured by the optimality gap. In order to evaluate the quality of the solutions we obtained from LS-B&B, we compare it against the optimal solution to the problem. It is however very difficult, if not impossible, to obtain and prove the optimal solution to the problems concerned. We therefore calculate the *approximate* optimal solution to the problem concerned by running the default B&B algorithm in CPLEX12.0 for an extensive amount of time.

In the comparison presented in Table 2, we aim to demonstrate the effectiveness of the repairing heuristic evoked in our proposed LS- B&B. Therefore, we present the characteristics of the sub-problems being repaired by heuristic against the characteristics of the default B&B. We compare LS-B&B with the default B&B in Table 2 in terms of the following criteria:

- The number of nodes being processed in B&B to obtain the best integer feasible solution;
- The gap between optimality and the quality of the best feasible solution;
- If the repairing heuristic is evoked and succeed;
- The total CPU time required.

Table 2. Comparisons of default B&B and LS-B&B. + denotes that the repairing heuristics are succeed. All the CPU time is measured in seconds.

		Société Générale	Hang Seng	DAX	FTSE	S&P	Nikkei
	No. of nodes processed	30	50	150200	147100	130800	35500
	Optimality Gap	0.22%	1.06%	4.66%	3.65%	2.74%	0.44%
	Repair success	No	No	No	No	No	No
	Total CPU time			180			
	No. of nodes processed	60	80	541800	486700	365800	105000
	Optimality Gap	0.1%	0.29%	4.66%	3.63%	2.74%	0.43%
Default B&B (original problem)	Repair success	No	No	No	No	No	No
	Total CPU time			600			
	No. of nodes processed	0+	0+	50+	30+	30+	50+
	Repair success	Yes	Yes	Yes	Yes	Yes	Yes
LS-B&B (sub-problem)	Optimality gap*	0.22%	1.07%	4.65%	3.67%	2.75%	0.44%
	Total CPU time *	3.16	3.09	9.00	11.44	13.55	76.97

In Table 2, in LS-B&B, the number of nodes processed is the average of nodes being processed with repairing heuristics. From Table 2 we can see that by simplifying the problem through variable fixing, the repairing heuristics succeed in LS- B&B approach. The repairing heuristics cannot be evoked by the default B&B while solving the original problem.

Without the simplification, the default B&B needs to explore a much larger number of nodes in the search to obtain feasible solutions, while LS-B&B with simplification requires much less time, shown in Table 2. For example, for the largest instance Nikkei, more than 35,500 nodes have been explored in the default B&B to obtain a feasible solution with a gap of 0.44%.

The optimality gap of solution obtained by LS- B&B is calculated by gap = $(f_{LS} - f_R) / f_R$, where f_{LS} is the objective value obtained by LS- B&B, and f_R is the objective value of continuous relaxation. Table 2 shows that, to achieve solutions of similar quality (as measured by the optimality gap), the CPU time needed by the default B&B is much greater than that required by LS-B&B (e.g. 180 CPU seconds as opposed to 76.97 seconds for the instance Nikkie).

The comparison of LS-B&B with the default B&B can be more clearly illustrated in Fig. 3, which plots of the objective values of LS-B&B and the approximate optimal values obtained by the default B&B with extensive runtime.

It can be seen that LS-B&B converges very well for instances Société Générale, Hang Seng and Nikkei, where the gap between the objective values of LS-B&B and approximate optimal is very small. For instance DAX, the best solution of LS-B&B is even better than the approximate optimal value. For instances FTSE and S&P, the gap is slightly larger. However, it should be noted that LS-B&B spends significantly less time (3-79 seconds) than the default B&B (180 and 600 seconds).

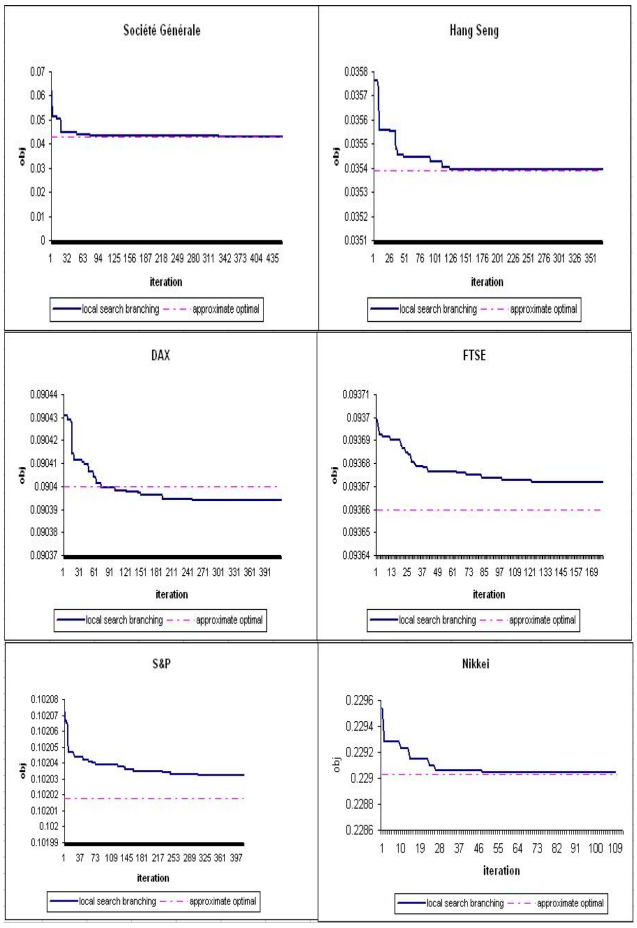

Fig. 3 The gap between LS-B&B and the approximate optimal by the default B&B

CONCLUSIONS

In this paper, we have introduced the hybrid LS-B&B method to solve the portfolio selection problem with practical trading constraints and transaction costs. We have analysed a specific PSP problem which is modelled as MIQP. The hybrid method closely integrates local search with B&B. It implements an incomplete search which aims to seek near optimal solutions in a limited computational time. It simplifies the problem into much smaller sub-problems, which are much easier to solve than the original complete problem, hence can be searched intensively by B&B. It has been demonstrated by our experiments that the repairing heuristics are evoked by solution information reusing in solving sub-problems, thus the successive sub-problems can be solved more efficiently. The heuristic initialization of the core variables in our problem provides a tight upper bound to prune more sub-problems.

REFERENCES

Bixby, R., M. Fenelon, Z. Gu, E. Rothberg and R. Wunderling (2000). MIP:Theory and practice--closing the gap. **System Modelling and Optimization: Methods,Theory and Applications 174**: 19-49.

Bienstock, D. (1996). Computational study of a family of mixed-integer quadratic programming problems. Mathematical Programming 74(2): 121-140.

Jobst, N. J., M. D. Horniman, C. A. Lucas and G. Mitra (2001). Computational aspects of alternative portfolio selection models in the presence of discrete asset choice constraints Quantitative Finance 1(5): 489-501.

Hansen, P., N. Mladenovic and D. Urosevic (2001). Variable neighborhood search: Principles and applications. European Journal of Operational Research 130(3): 449-467.

He, F. and R. Qu, A two-stage stochastic mixed-integer program modelling and hybrid solution approach to portfolio selection problems. Information Sciences, 289: 190–205, 2014.

Konno, H. and A. Wijayanayake (2001). Portfolio optimization problem under concave transaction costs and minimal transaction unit constraints. Mathematical Programming 89(2): 233-250.

Konno, H. and A. Wijayanayake (2002). Portfolio optimization under D.C. transaction costs and minimal transaction unit constraints. Journal of Global Optimization 22(1): 137-154.

Lazic, J., S. Hanafi, N. Mladenovi and D. Urosevic (2009). Variable neighbourhood decomposition search for 0-1 mixed integer programs. Computers &Operations Research 37(6): 1055-1067.

Mansini, R. and M. G. Speranza (1999). Heuristic algorithms for the portfolio selection problem with minimum transaction lots. European Journal of Operational Research 114(2): 219-233.

Stoyan, S. and R. Kwon (2010). A two-stage stochastic mixed-integer programming approach to the index tracking problem. Optimization and Engineering 11(2): 247-275.

Stoyan, S. J. and R. H. Kwon (2011). A Stochastic-Goal Mixed-Integer Programming approach for integrated stock and bond portfolio optimization. Comput. Ind. Eng. 61(4): 1285-1295.

AUTHOR BIOGRAPHIES

Fang He was born in China and obtained her PhD degree from The University of Nottingham, U.K. And she worked as a Research Fellow at the same univerity for 3 years on modelling and optimisation for combinatorial optmisation problem in real-world applications. This work is condunt during that time. Now she is a lecturer in the Department of Computer Scicence, Univeristy of Westminster.

Rong Qu is an Associate Professor of Computer Science, at the School of Computer Science, The University of Nottingham, U.K. Her research interests are on the modelling and optimisation algorithms (meta-heuristics, mathematical approaches and their hybridisations) to real-world optimisation and scheduling problems.