Abstract—Single carrier frequency division multiple access (SC-FDMA) has been adopted as the standard multiple access scheme for 3GPP LTE uplink. In comparison to orthogonal frequency division multiple access (OFDMA), the subcarriers assigned to each user are required to be consecutive in SC-FDMA localized scheme, which imposes more difficulties on resource allocation problem. Subject to this constraint, various optimization objectives, such as utility maximization and power minimization, have been studied for SC-FDMA resource allocation. In this paper, we focus on developing a general algorithm framework with near-optimal performance and polynomial-time complexity to maximize the total utility for SC-FDMA systems. The proposed algorithm is based on low-complexity estimation for the partial solution space. Compared with existing algorithms, simulation results show that our algorithm improves the system utility significantly and has less deviation to global optimum. In addition, the proposed algorithm framework allows a flexible trade-off between computational effort and solution performance by varying the complexity of estimation approaches.

Index Terms—SC-FDMA; Algorithm; Resource Allocation; Binary Integer Programming; Partial Solution; Estimation.

I. INTRODUCTION

Orthogonal frequency division multiple access (OFDMA) is known to be a promising technology to offer high-speed data transmission for future wireless communication systems such as The Third Generation Partnership Project Long Term Evolution (3GPP-LTE). However, one disadvantage of OFDMA is high peak-to-average-power ratio (PAPR) which requires more complicated amplifier design and thus higher power consumption in user equipment. Relative to OFDMA, in single carrier frequency division multiple access (SC-FDMA), each subcarrier is spread in the frequency band, and subcarriers are transmitted sequentially rather than in a parallel manner in time domain as in OFDMA, which reduces the envelope fluctuations in the transmitted waveform and offers lower PAPR. Thus, as an alternative to OFDMA, SC-FDMA has been adopted as the uplink multiple access scheme in 3GPP-LTE standard [1]. SC-FDMA has the similar structure to OFDMA. For both multiple access schemes, the entire frequency bandwidth is divided into a large number of orthogonal narrow-band subcarriers to be dynamically shared by multiple users. Assigning subcarriers to the more favorable users can provide beneficial diversity to the system.

Most of the previous work has focused on subcarriers and power allocation in OFDMA system. Many algorithms have been proposed for OFDMA resource allocation with the objective of maximizing the utility or minimizing the power consumption [3]–[6]. In SC-FDMA, there are two subcarrier mapping schemes, localized and interleaved allocation [2]. In localized SC-FDMA, subcarriers assigned to each user required to be consecutive in the frequency domain. From the standpoint of optimization, this constraint is the major difference between SC-FDMA and OFDMA. Such a restriction makes the optimization problem in SC-FDMA even harder to solve, and the proposed algorithms for downlink OFDMA can not be directly applied to uplink SC-FDMA.

For SC-FDMA, complexity analysis under the subcarrier-consecutive constraint has been studied in [7][10]. In [10], the authors provide the NP-hardness proof for the scheduling problem of maximizing the sum utility. In [9], a throughput maximization problem in SC-FDMA has been considered where the problem was formulated as a pure binary-integer programming (BIP). This allows the optimal solution to be computed with reasonable complexity compared to an exhaustive search, but it is still not applicable for practical systems. Thus, some sub-optimal heuristic algorithms have been proposed for SC-FDMA resource allocation. In [11], a so called mean-enhanced algorithm has been proposed for dynamic subcarrier allocation. The subcarriers-consecutive allocation is based on users’ average channel gains. In [10], the authors propose several greedy-based algorithms considering proportional fair (PF). In their algorithms, a subcarrier is assigned to the user with the highest channel gain iteratively until all the subcarriers are assigned. A heuristic algorithm based on steepest ascent has been presented in [9]. In this algorithm, exactly one subcarrier is allocated to one user at each iteration and the subcarrier-user allocation is determined by the maximum throughput increase. In [8] the authors extended SC-FDMA allocation problem described in [9] by considering minimizing total power and the number of used channels with subcarrier-consecutive constraint. Some greedy algorithms also have been proposed for SC-FDMA systems in their work.

However, pure heuristics such as conventional greedy algorithms have a myopic view of the whole solution space. To avoid being stuck in a local domain of solution space, it is desirable to design an algorithm that has a more global view of the whole solution space and thus provides better solution.

In this paper, we consider sum utility maximization under the subcarrier-consecutive constraint in SC-FDMA systems. Our objective is to develop a polynomial-time complexity algorithm with high competitiveness in approaching global optimum. Firstly, we formulated the problem as a binary integer programming to attain the optimal solution. Then, we proposed a partial solution estimation based algorithm (PSEA)
to maximize the sum utility. The proposed algorithm not only runs in its basic version with low complexity, but also adopts sub-algorithms to estimate the solution for the unallocated resource in each iteration. Thus, PSEA have an overall view of the whole solution space. Numerical results demonstrate PSEA is able to provide a close-to-optimal solution.

The contributions are as follows: 1) The algorithm (PSEA) exhibits promising performance in approaching global optimum with increase of estimation accuracy. 2) By varying the computational effort of estimation, PSEA provides a flexible trade-off between solution quality and computational complexity. 3) We show that the PSEA has the global optimality guarantee for certain cases. 4) We demonstrate a competitive performance of PSEA compared to existing algorithms.

The paper is organized as follows. Section II describes the system model and the problem definition. Section III presents the proposed algorithm and analyzes the computational complexity. In Section IV, we present the numerical results of performance evaluation and finally Section V summarizes the key conclusions.

II. SYSTEM MODEL AND PROBLEM DEFINITIONS

We consider an uplink SC-FDMA multiuser system with one base station. The overall bandwidth is divided into $N$ narrow-band subcarriers to be allocated to $K$ users. We use $\mathcal{K} = \{1, \ldots, k, \ldots, K\}$ and $\mathcal{N} = \{1, \ldots, n, \ldots, N\}$ to denote the sets of users and subcarriers, respectively. We consider subcarriers independently undergo Rayleigh fading. The channel gain-to-noise ratio for user $k$ on subcarrier $n$ is denoted by $\gamma_{k,n}$.

In SC-FDMA, the subcarriers assigned to each user must be consecutive in localized allocation scheme. With $N$ subcarriers, it is straightforward to see that the number of all the possible subcarrier blocks is $\mathcal{N}(N+1)/2$. We denote $\mathcal{B} = \{1, \ldots, b, \ldots, B\}$ as the set of these subcarrier blocks where $B = \mathcal{N}(N+1)/2$. Each element $b$ consists number of consecutive subcarriers. Due to the consecutive constraint, any adjacent subcarriers can be grouped into one block. We treat such subcarrier block as the basic resource allocation unit in the frequency domain.

In terms of power allocation, the transmit power for each user $k$ should be less than a power level limit, denoted by $P_{k}^{\max}$ and the power on the subcarriers assigned to a user should be equal, i.e. $\frac{P_{k}^{\max}}{|b|}$. In addition, for each individual subcarrier $n$, the power can not exceed a given peak power level $P_{n}^{\max}$.

Let $v_{k,b}$ denote the utility value of subcarrier block $b$ allocated to user $k$. Typically, the utility can be considered as the data rate achieved by allocating subcarrier block $b$ to user $k$ with power $\min\{\frac{P_{k}^{\max}}{|b|}, P_{n}^{\max}\}$ on each subcarrier, as well as introducing scaling factors to reflect user’s priority or balance the fairness among users. For simplicity, the sum utility is equivalent to system throughput in this work. However, the proposed algorithm is not restricted in this. The Shannon’s capacity function is used for achievable rate computation. The utility of allocating subcarrier block $b$ to user $k$ can be expressed as:

$$v_{k,b} = \sum_{n \in b} \log_2(1 + \min\{\frac{P_{k}^{\max}}{|b|}, P_{n}^{\max}\} \gamma_{k,n})$$ (1)

In order to obtain the optimal solution with reasonable complexity compared to an exhaustive enumeration, we formulate this optimization problem by means of the following binary integer programming (BIP) model.

We denote a binary variable $x_{k,b}$ which is 1 if subcarrier block $b$ is assigned to user $k$, otherwise 0. Let $b_{k}$ denote subcarrier block $b$ allocated to user $k$ in the resource allocation solution. The objective of the integer programming model is to maximize the sum utility:

maximize: $\sum_{k=1}^{K} \sum_{b=1}^{B} x_{k,b} v_{k,b}$ (2)

s.t.: $\sum_{b=1}^{B} x_{k,b} \leq 1, \ \forall k \in K$, (3)

$$b_{k} \cap b_{k'} = \emptyset, \ \forall k \neq k', \ \forall k, k' \in \mathcal{K}.$$ (4)

In the formulation, (2) shows the objective of maximizing the total utility. Constraint (3) represents that one user can select at most one subcarrier block. Constraint (4) is the subcarrier exclusivity restriction. There is no subcarrier (or subcarrier block) reuse among users to avoid interference.

In this optimization problem, the binary variable turns the problem into combinatorial domain. The combinatorial nature stems from two facts: subcarrier exclusivity and subcarrier consecutive constraint. In general, using state-of-the-art software to solve such combinatorial problems is considered not very practical. However, it provides the global solution which can be viewed as a benchmark for the performance evaluation of the suboptimal algorithms.

III. PROPOSED ALGORITHM

Due to the NP-hardness of the utility maximization problem, it is impossible to develop a polynomial-time algorithm to obtain the optimal solution, unless $P = NP$ [10]. Although exhaustive search algorithms always produce the optimal solution [12], but usually it is prohibitive in practice because of high computational complexity. Therefore, it is preferable to design an effective algorithm with polynomial-time though it maybe suboptimal.

In this section, we propose a so called partial solution estimation based algorithm (PSEA). We first present the framework of PSEA. Afterwards, based on this basic form, we propose an estimation strategy to enhance the algorithm.

A. Algorithm Framework

In the algorithm design, we adopt part of the concept of dynamic programming. It is known that dynamic programming is an effective technique for implementing a recursive algorithm by storing partial results [12]. The main idea of our proposed algorithm framework is: 1) divide the original problem into
multi-stage subproblems; 2) systematically search all the possible user-block allocation patterns; 3) store the best partial solution at each stage to avoid recomputing. The maximum utility is the accumulation of the local solutions in the previous stages.

For an optimization problem, if the locally optimal solutions of each sub-problems are themselves part of the globally optimal solution, then say this problem has the optimality substructure [12]. In SC-FDMA, each user can occupy at most one subcarrier block and each subcarrier can be used at most once. These constraints result in the user-block decision problems no longer having the optimal substructure because the decision of current stage will affect the decision of later stages. Therefore, in general, the proposed algorithm returns the suboptimal solution.

We divide the whole decision procedure into \{1, \ldots, n, \ldots, N\} multiple stages. The number of decision stages equals the number of subcarriers. At each stage \(n\), we solve the sub-problem of allocating the first \(n\) subcarriers and \(|\mathcal{K}_n|\) candidate users, where \(\mathcal{K}_n \subseteq \mathcal{K}\).

The computations are carried out in a forward procedure, processing from stage 1 to stage \(N\). We solve the subproblem sequentially and store the best possible solution \(S^*(n) = (u^*_n, \mathcal{K}^*_n)\) at each stage \(n\). \(S(n)\) is a tuple consisting of the accumulated utility \(u_n\) and \(\mathcal{K}_n\) is the associated subset of users for which subcarrier blocks have been assigned. Thus, the best partial solution \(S^*(n)\) can be reused later.

In order to find \(S^*(n)\) at stage \(n\), we exploit all the possible subcarrier block-user allocation. The utility of assigning one of the candidate subcarrier blocks to one additional user is correspondingly added to the stored maximum utility in previous stage. For example, at stage \(n\), using the stored utility \(u^*_n\) in \(S^*(i), i < n\) and allocating candidate consecutive subcarriers \(i + 1, \ldots, n\) to an additional user \(k\), is one of the feasible solutions. The resulting new solution is given by \((u^*_{i+1,n} + u^{k}_{i+1,n} \cup \{k\})\), where \(u^k_{i+1,n}\) is the utility of assigning consecutive subcarriers \(i + 1, \ldots, n\) to user \(k\). The maximum utility \(u^*_n\) for each stage can be computed by evaluating all the possible user-block allocation patterns in the following formulation (5).

\[
\begin{align*}
0 + u^k_{1,n}, & \quad \forall k \in \mathcal{K}, \text{ allocate subcarriers} \\
1, \ldots, n \text{ to user } k \\
u^*_n + u^k_{j,n}, & \quad \forall k \in \mathcal{K} \setminus \mathcal{K}^*_j, \text{ allocate subcarriers} \\
2, \ldots, n \text{ to user } k \\
\vdots & \\
u^*_{n-1} + u^k_{n}, & \quad \forall k \in \mathcal{K} \setminus \mathcal{K}^*_{n-1}, \text{ allocate subcarriers} \\
n \text{ to user } k
\end{align*}
\]

After traversing all the possible solutions, the best partial solution \((u^*_n, \mathcal{K}^*_n)\) is stored in \(S^*(n)\). Once the processing is completed at the end of stage \(N\), the solution \(S^*(N)\) gives the overall maximum utility.

In some cases, allocating a subcarrier block to a user may not always be optimal. Although locally allocating a subcarrier block to a user is always equal or better than not assigning to any user, but globally, skipping the allocation for a subcarrier block may potentially improve the overall solution, e.g., when the channel conditions on these subcarriers are poor. We take this aspect into account in implementation. Consider a possible solution in stage \(n\) to be derived from \(S^*(j)\) with \(j < n\), the set of candidate users for subcarrier block \(j + 1, \ldots, n\) is \(\mathcal{K} \setminus \mathcal{K}^*_j\) which is the users that have not been allocated any subcarrier blocks in \(\mathcal{K}^*_j\). Therefore, there are \(|\mathcal{K} \setminus \mathcal{K}^*_j| + 1\) possible user choices for subcarrier block \(j + 1, \ldots, n\) in the algorithm implementation. Among them, one is not to assign any user to this block at all.

Instead of simply picking and storing the maximum utility at each decision stage, the algorithm framework adopts some rules to avoid storing the maximum utility which locally has the best performance but evidently non-optimal. For example, at stage \(n\), consider two possible solutions denoted by \(S'(n) = (u'_n, \mathcal{K}'_n)\) and \(S''(n) = (u''_n, \mathcal{K}''_n)\). If \(u'_n\) is the maximum utility and greater than \(u''_n\), then obviously \(S'(n) = S''(n)\). But if both \(u'_n = u''_n\) yield the maximum utility and \(\mathcal{K}'_n \subset \mathcal{K}''_n\), then solution \(S'(n)\) will appear in \(S^*(n)\), since the later will not achieve a better overall solution. This fact is formalized as follows.

**Lemma 1.** Consider two possible solutions of allocating the first \(n\) subcarriers, denoted by \(S'(n) = (u'_n, \mathcal{K}'_n)\) and \(S''(n) = (u''_n, \mathcal{K}''_n)\). If the maximum utility \(u'_n = u''_n\) and \(\mathcal{K}'_n \subset \mathcal{K}''_n\), then \(S'(n)\) performs over \(S''(n)\) and thus \(S^*(n) = S'(n)\), that is, for any solution derived by the later, there exists a solution enabled by the former with equal or even better performance.

**Proof.** Assume \(u'_n = u''_n\) and \(\mathcal{K}'_n \subset \mathcal{K}''_n\). In user set \(\mathcal{K} \setminus \mathcal{K}'_n\), there exists at least one user \(l \notin \mathcal{K} \setminus \mathcal{K}'_n\), but \(l \in \mathcal{K} \setminus \mathcal{K}'_n\). Thus, at the final stage \(N\), the overall utility \(\max(u'_n + u''_{j+1,N})\), where \(k \in \mathcal{K} \setminus \mathcal{K}'_n\), derived from \(S''(n)\) with allocation of remaining subcarriers \(j + 1, \ldots, N\), has equal or better performance than \(\max(u''_n + u''_{j+1,N})\), where \(h \in \mathcal{K} \setminus \mathcal{K}'_n\). For example, user \(l\) is the best choice for remaining subcarriers \(j + 1, \ldots, N\).

**B. The PSEA Algorithm**

In the proposed algorithm framework, the key idea is to store the best partial solution at each decision stage. However, this strategy may not always lead to a better overall solution in some cases, because a partial solution which is globally optimal may not necessary locally have good performance. For example, allocating subcarrier block \(b\) to user \(i\) may be globally optimal, but locally with poor performance. If an assignment to another user \(j\) leads better performance, then the former optimal solution has no chance to be stored in the current \(S^*(n)\) at all and can not be reused later. To tackle this issue, we propose an estimation strategy based on the algorithm framework.

Instead of only computing and storing the utility of partial solution, PSEA also estimates the utility for the unallocated resource by using a sub-algorithm. Thus, at each decision stage we have a more global view for the whole solution space.

We use \(E(N_e, \mathcal{K}_e)\) to denote the utility estimation for the remaining subcarrier and user set, denoted by \(N_e\) and \(\mathcal{K}_e\), respectively.
The proposed estimation strategy will lead to a better overall performance. Consider at stage \( n \), allocating subcarrier block \( i + 1, \ldots, n \) to user \( k \) is globally optimal, but \( u_{i+1,n}^k \) locally has poor performance. Then, the solution \( S(n) = (u_i^k + u_{i+1,n}^k, \mathcal{K}_i^* \cup \{k\}) \), \( i \leq n \) will definitely fail to appear in \( S^*(n) \), which may degrade the overall performance. However, the computing approach becomes \( S'_n(n) = (u_i^k + u_{i+1,n}^k + E(\mathcal{N}_e, \mathcal{K}_e), \mathcal{K}_i^* \cup \{k\}) \) in PSEA, where \( \mathcal{N}_e = \{n+1, \ldots, N\} \), \( \mathcal{K}_e = \mathcal{K} \setminus (\mathcal{K}_i^* \cup k) \). If \( S(n) \) yields the best solution at current decision stage \( n \), then \( S^*(n) = (u_i^k + u_{i+1,n}^k, \mathcal{K}_i^* \cup \{k\}) \). Therefore, for any part of optimal allocation, e.g. \( u_{i+1,n}^k \), the estimation strategy provides more opportunity to store partially optimal solution in the \( S^*(n) \), and thus improves the overall performance later.

By adopting different sub-algorithms to compute \( E(\mathcal{N}_e, \mathcal{K}_e) \), e.g. any existing low-complexity greedy algorithm, PSEA offers a flexible trade-off between computational complexity and performance. Moreover, if the estimation is accurate in PSEA, then the global optimum is achieved, even at the first decision stage. This fact is formalized in the lemma below.

**Lemma 2.** If the utility estimation \( E(\mathcal{N}_e, \mathcal{K}_e) \) is accurate in PSEA, then any of the best partial solutions \( S^*(n) \) is globally optimal and PSEA guarantees global optimality.

**Proof.** Assume estimation \( E \) returns optimal solution for the remaining resource \( (\mathcal{N}_e, \mathcal{K}_e) \). At stage 1, there exists an optimal user-block allocation for subcarrier block 1, i.e. \( u_1^k, k^* \in \mathcal{K} \), where \( k^* \) is the optimal user choice for subcarrier block 1, then \( u_1^k + E(\mathcal{N}_e, \mathcal{K}_e) \) is globally optimal and \( u_1^k \) will be stored in \( S^*(1) \). The process for the later stages \( n > 1 \) follows analogously. For any subcarrier blocks \( i + 1, \ldots, n, i < n \), deriving from the previous stage \( u_i^k \), the utility \( u_i^k + E(\mathcal{N}_e, \mathcal{K}_e) \), where \( \mathcal{N}_e = \{n+1, \ldots, N\} \), \( \mathcal{K}_e = \mathcal{K} \setminus (\mathcal{K}_i^* \cup k^*) \), will yield the best performance and thus be stored in \( S^*(n) \), which implies at each stage, \( S^*(n) \) is also globally optimal solution. Thus, PSEA guarantees global optimality.

The computation flow of PSEA is formalized in Algorithm 1. In Line 2, the solution for each subproblem \( S^*(j) \) is set to \( \{0, \emptyset\} \) at the initialization step. As mentioned in Section III, there are \( |\mathcal{K} \setminus \mathcal{K}_j^*| + 1 \) possible choices for block allocation, so we add an virtual user 0 with no utility and keep user 0 always available for any possible allocation to skip some block allocation in algorithm implementation.

The algorithm starts at Line 3. The bulk of PSEA processes stages from 1 to \( N \), one by one. For each stage, in Line 4 to Line 11, PSEA collects the stored solution \( u_i^k \) in previous stage, evaluates all the possible solutions and temporarily stores in \( S' \) (Line 6), picks the best possible solution from \( S' \), which satisfies the rules in Lemma 1 (Line 9 to Line 10), then stores this solution in \( S^*(j) \). The algorithm terminates when the treatment of stage \( N \) is completed. Then \( u_N^k \) yields the best performance value, which represents the algorithm outcome.

**Algorithm 1** The partial solution estimation based algorithm (PSEA)

**Input:** \( \mathcal{K} = \{0, \ldots, k, \ldots, K\} \), \( v_{k,b} \) (the utility of block \( b \) allocated to user \( k \))

**Output:** \( u_N^k \) (the maximum utility in stage \( N \))

1: for \( j = 0 : N \) do
2: \( S^*(j) = \{0, \emptyset\} \)
3: for \( j = 1 : N \) do
4: for \( i = 0 : j - 1 \) do
5: for all \( k \in \mathcal{K} \setminus \mathcal{K}_i^* \) do
6: \( S' \leftarrow S' \cup \{u_i^k + u_{i+1,j}^k + E(\mathcal{N}_e, \mathcal{K}_e), \mathcal{K}_i^* \cup \{k\}\} \)
7: end for
8: end for
9: \( u_i^j \leftarrow \arg\max \{u_i^k + u_{i+1,j}^k + E(\mathcal{N}_e, \mathcal{K}_e)\} \) in \( S' \)
10: \( \mathcal{K}_j^* \leftarrow \mathcal{K}_i^* \cup \{k^*\} \)
11: \( S^*(j) \leftarrow (u_i^j, \mathcal{K}_i^* \cup \{k^*\} \)
12: end for
13: return \( u_N^k \)

**C. The Algorithm Complexity**

We first analyze the complexity of the algorithm framework (without estimation \( E(\mathcal{N}_e, \mathcal{K}_e) \)). There are clearly no more than \( N^2 \) iterations in Line 3 and Line 4. In Line 5, the worst case is that the best user choice is always selecting use 0 (user 0 can be reused) in every stage. Thus, the loop in Line 5 requires \( |\mathcal{K}| \) iterations. Next, in Line 6 each operation requires \( O(1) \) computing time. Therefore, the worst case complexity of algorithm framework is \( O(N^2 K) \). With estimation \( E(\mathcal{N}_e, \mathcal{K}_e) \), the overall complexity of PSEA in the worst case is \( O(N^2 K + O(E(\mathcal{N}_e, \mathcal{K}_e))) \).

**IV. Performance Evaluation**

We evaluate the performance of the proposed PSEA algorithm. We consider the SC-FDMA uplink of a LTE cell with randomly distributed users.

Some key parameters are summarized in Table I. We use four sets of data with \( (K, N) = (10, 64), (20, 64), (10, 128) \) and \( (20, 128) \), to investigate algorithms’ performance. Each value in the following results represents the average performance over 100 instances.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cell radius</td>
<td>1000 m</td>
</tr>
<tr>
<td>Carrier frequency</td>
<td>2000 MHz</td>
</tr>
<tr>
<td>Number of subcarriers</td>
<td>64, 128</td>
</tr>
<tr>
<td>Subcarrier bandwidth</td>
<td>15 KHz</td>
</tr>
<tr>
<td>User power constraint</td>
<td>200 mW</td>
</tr>
<tr>
<td>Subcarrier power constraint</td>
<td>10 mW</td>
</tr>
<tr>
<td>Power spectral density of noise</td>
<td>-174 dBm/Hz</td>
</tr>
<tr>
<td>Path loss</td>
<td>COST-231-HATA</td>
</tr>
<tr>
<td>Shadowing</td>
<td>Log-normal, 8 dB standard deviation</td>
</tr>
<tr>
<td>Multipath fading</td>
<td>Rayleigh fading</td>
</tr>
</tbody>
</table>

To obtain global optimum, the integer programming model in Section II is implemented in AMPL [13] and solved by CPLEX [14]. Two greedy algorithms, Maximum-Utility-Increase (MUI) algorithm [8] and Riding Peak (RP) algorithm...
[10], have been implemented for performance comparison. Note that, the objective of original RP algorithm [10] focuses on proportional fair (PF) maximization. We change this objective to system throughput maximization in the simulation.

In Fig. 1, we show the algorithm’s performance of the average optimality gap. Throughout this section, the term optimality gap refers to the relative difference in comparison to the global optimum. In Fig. 1, algorithm “PSEAF” represents the algorithm framework of PSEA proposed in Section III without estimation strategy. “PSEA with E(MUI)” and “PSEA with E(RP)” mean algorithm PSEA adopting MUI and RP algorithm as the estimation approach, respectively. Then, we make three following observations based on Fig. 1:

First, although the proposed PSEAF (green bar) has the same complexity with algorithm MUI and near-complexity with RP, PSEAF still provides better performance than MUI and RP. The worst case complexity of MUI is $O(N^2K)$ [9], as same as PSEAF. From Fig 1, the optimality gaps of MUI and RP range from approximately 17% to 24%, and 21% to 27%, respectively. PSEAF yields much better results than MUI and RP, ranging between 6% and 17% in global optimality gaps.

Second, the proposed estimation strategy in PSEA results in significant performance improvement. As can be seen from Fig 1, the optimality gaps in both PSEA with $E(MUI)$ and PSEA with $E(RP)$ are sharply reduced with range being approximately 2.5% and 7%. It indicates that PSEA not only significantly improves the performance of greedy based algorithm MUI and RP, respectively, but also enhances the algorithm framework (PSEAF).

Third, empirically, the PSEA with higher quality of estimation approach offers better overall solution. Compared with algorithm RP, although MUI gives moderate improvement, MUI based estimation can be viewed as more accurate than RP based estimation. Numerically, PSEA with $E(MUI)$ exhibits more competitive performance than PSEA with $E(RP)$.

VI. ACKNOWLEDGEMENTS

This work has been supported in part by the EC Marie Curie projects MESH-WISE (FP7-People-2012-IAPP: 324515), WiNDOWN (FP7-People-2012-IAPP: 318992), and the Linköping-Lund Excellence Center in Information Technology (ELLIIT), Sweden. The work of the first author is supported by the China Scholarship Council (CSC). The second author is partially supported by EC-FP7 Marie Curie CIG grant with proposal number 294182.

REFERENCES