Vis Comput (2010) 26: 287–292
DOI 10.1007/s00371-009-0406-5

ORIGINAL ARTICLE

Some notes on maximal arc intersection of spherical polygons: its $\mathcal{NP}$-hardness and approximation algorithms

Yong-Jin Liu · Wen-Qi Zhang · Kai Tang

Published online: 1 December 2009

Abstract Finding a sequence of workpiece orientations such that the number of setups is minimized is an important optimization problem in manufacturing industry. In this paper we present some interesting notes on this optimal workpiece setup problem. These notes show that (1) The greedy algorithm proposed in Comput. Aided Des. 35 (2003), pp. 1269–1285 for the optimal workpiece setup problem has the performance ratio bounded by $O(\ln n - \ln\ln n + 0.78)$, where n is the number of spherical polygons in the ground set; (2) In addition to greedy heuristic, linear programming can also be used as a near-optimal approximation solution; (3) The performance ratio by linear programming is shown to be tighter than that of greedy heuristic in some cases.

Keywords Spherical polygons intersection · NC machining · $\mathcal{NP}$-hard problem · Approximation algorithms

1 Introduction

In industrial mass-production systems, the time to mount, calibrate and dismount the workpieces could take considerable time in comparison to the actual machining time. More critically, different mounting of a workpiece always introduces precision errors in machining. Accordingly finding a sequence of workpiece orientations such that the number of setups is minimized is an important optimization problem in CAD/CAM.

In 1992 Tang et al. [19] proposed a computational model with a greedy heuristic to solve the optimal workpiece setup problem for 4- and 5-axis numerically-controlled (NC) machining. Since the underlying geometric model is the same for both 4- and 5-axis machining, in the following discussion we focus on the geometric model of 4-axis machining.

As illustrated in Fig. 1, in a 4-axis NC machine, the cutter can move up and down in the y direction, while the worktable can be translated back and forth in the x and z directions, together with an additional rotation about the x direction. Assume a ball-end cutter is used. The range of a cutter to access a workpiece surface can be characterized by its visibility map on a Gauss sphere [7, 18]. With the considerations of global accessibility and interference, the visibility maps can be any general spherical polygons (not only convex as assumed in [4]). By representing a mechanical part by a group of machinable surfaces, the workpiece accessibility can be characterized by a set of (possibly overlapped) spherical polygons on a Gauss sphere $\mathbb{S}^2$ (ref. Fig. 2).

In a single setup with a 4-axis NC machine, the directions along which the cutter can access the workpiece can be represented by an arc $\theta \leq 2\pi$ on $\mathbb{S}^2$ (ref. Fig. 2). Tang et al. [19, 20] formulated the optimal workpiece setup problem as the following geometric optimization problem:

Problem 1 Given a set P of spherical polygons, each of which corresponds to accessible orientations for the cutter to a subset of grouped surfaces in workpiece, find a set A of arcs of a given length $\theta \leq 2\pi$ with a minimal cardinality such that every spherical polygon intersects at least one of the arcs in A.

Y.-J. Liu (✉) · W.-Q. Zhang
Department of Computer Science and Technology, Tsinghua National Lab for Information Science and Technology, Tsinghua University, Beijing, P.R. China
e-mail: liuyongjin@tsinghua.edu.cn

K. Tang
Department of Mechanical Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong

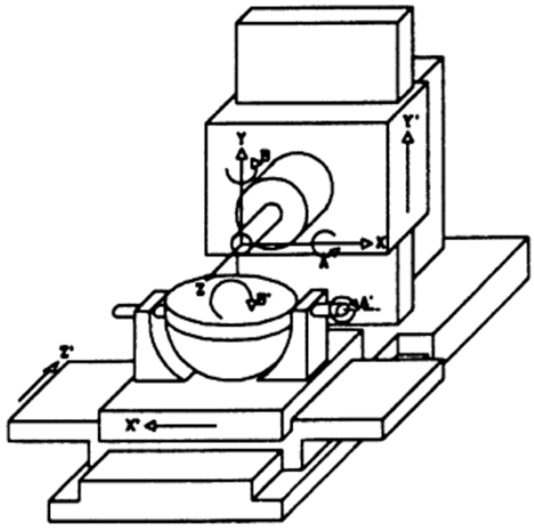

Fig. 1 4-axis milling machine

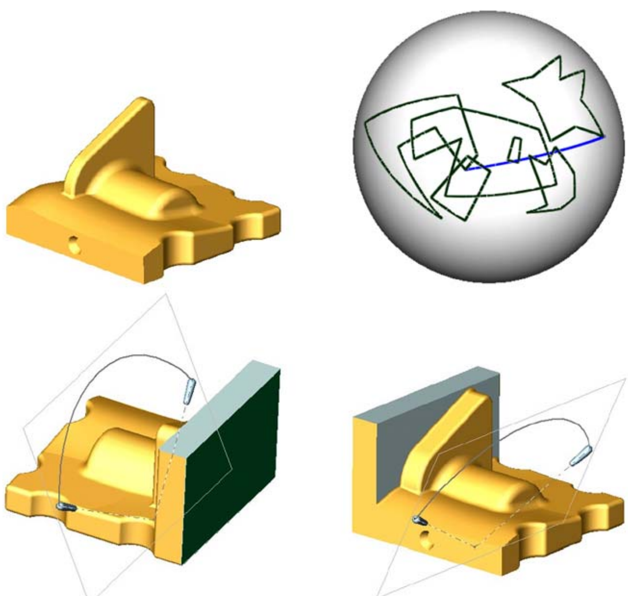

Fig. 2 A snapshot of implementation of the approximation algorithm [16] to solve Problem 2

Tang et al. [19] also first proposed to solve Problem 1 by a greedy heuristic with an algorithmic solution to the following problem:

Problem 2 Given a set P of spherical polygons, find an arc of length $\theta \leq 2\pi$ that intersects a maximal number of polygons in P.

In other words, the greedy approach is that at each step, a maximal cutting arc is computed and the spherical polygons intersected by it are removed from the input set of polygons. This process is repeated until the polygon set becomes empty.

In the last decade, a number of optimization algorithms [4, 9, 19, 20] have been proposed to solve some special cases (with increased generality) of Problem 2. Recently, the most general case of Problem 2 has been solved in [16]. Two similar cases in a much broader ground are also discussed in [1, 17]. Despite its solvability of Problem 1 via Problem 2, one may ask, "Can Problem 1 be solved directly by any optimization algorithm?" or "Is there any better approximation algorithm than the greedy one?"

In this paper we present in Sect. 3 that Problem 1 is $\mathcal{NP}$-hard. Although this result is fairly known in theoretic computer science, we believe that explicit proof of this result gives some guidelines in CAD/CAM community: by knowing this result, we can answer the first question since the $\mathcal{NP}$-hardness of a problem offers good evidence for its intractability. We answer the second question in Sect. 4 by showing that linear programming can also be used as an alternative approximation solution and its performance ratio is better than that of the greedy algorithm in some cases.

2 Notations

The following classic problems are cited in this paper.

Problem 3 (Vertex cover problem) Given a graph G and an integer k, is there a vertex cover[1] of cardinality k?

Problem 4 (Set cover problem) Given a universe U of n elements and a collection S of subsets of U, find a set cover of the set system (U, S) with a minimal cardinality, i.e., a collection of minimal sets $\widetilde{S} \subseteq S$ such that $\bigcup_{s \in \widetilde{S}} s = U$.

It is well known [8] that Problem 3 is a decision problem and is $\mathcal{NP}$-complete, and Problem 4 is an optimization problem and is $\mathcal{NP}$-hard.

Informally any language $L \subseteq \{0, 1\}^*$ can be interpreted as a decision problem like Problem 3. For an optimization problem like Problem 4, to make it a decision problem, we simply ask whether or not a set cover with cardinality less than a given integer k exists. A language L_1 is *Karp-reducible* to a language L_2, denoted by $L_1 \preceq L_2$, if there exists a function f such that

1. f maps every instance of L_1 to an instance of L_2, and
2. f satisfies $x \in L_1$ if and only if $f(x) \in L_2$ for all x.

If the reduction function can be computed in polynomial time, L_1 is polynomial-time reducible to L_2, denoted by $L_1 \preceq_P L_2$. In the rest of this paper, we prove that Problem 3 $\preceq_P$ Problem 1 $\preceq$ Problem 4.

[1] A *vertex cover* of an undirected graph $G = (V, E)$ is a subset $V' \subseteq V$ such that if $(u, v) \in E$, then $u \in V'$ or $v \in V'$, or both.

3 Problem 1 is $\mathcal{NP}$-hard

We have the following result.

Note 1 *Problem 1 is $\mathcal{NP}$-hard.*

Proof We prove this result by showing that Problem 3 $\preceq_P$ Problem 1. The method we use is to show that Problem 3 is a special case of Problem 1.

The special case of Problem 1 can be constructed as follows. First if the arcs in Problem 1 have length $\theta = 0$, then the arcs degenerate to points. Now a set of spherical polygons intersected by an arc means this set of polygons is sharing a common point. Secondly, if the polygons in Problem 1 shrink into edges, then a set of polygons sharing a common point means a set of edges is sharing a common intersection point. Finally if we restrict the edges in the universe P of Problem 1 to only intersect each other at their endpoints, then the edges in P together with their endpoints form a graph G and now the Problem 1 is equivalent to the following one: given a graph $G = (V, E)$, find a subset $V' \subseteq V$ with minimal cardinality such that every edge $e \in E$ has at least one endpoint in V'. This is exactly the optimization version of Problem 3. □

To the best knowledge of the authors, there is no explicit statement of the above result that was published before.

Note 2 *Problem 1 $\preceq$ Problem 4, i.e., every instance of Problem 1 can be mapped to a corresponding instance of Problem 4 such that every near-optimal solution output from the greedy algorithm for an instance of Problem 1 is exactly the same near-optimal solution output from the greedy algorithm for the corresponding instance of Problem 4.*

Proof Problem 1 can be re-casted as follows. Let U be a finite set of spherical polygons with the cardinality $|U| = n$. The power set of U, denoted by $P(U)$, is the collection of all the subsets of U. Clearly $|P(U)| = 2^n$. Given an arc of length $\theta \le 2\pi$, denote the subset of $P(U)$ by S which satisfies $S = \{s_i \mid s_i \in P(U)$ and all polygons in s_i can be intersected by an arc of a given length $\theta\}$. Let $|S| = m$. Obviously $S = \{s_1, \ldots, s_m\}$ is a cover of U, i.e., $\bigcup_{i=1}^{m} s_i = U$. Note that in this reduction the mapping function can take worst-case super-polynomial time $O(2^n)$, since $|P(U)| = 2^n$.

Now it is clear that solving an instance of Problem 1 is equivalent to solving the corresponding instance of Problem 4 and, the algorithm `maximal-arc-intersection` to solve Problem 2 is equivalent to the operation 2.1 in the `greedy-set-cover` algorithm (presented in Sect. 4.1). Hence every near-optimal solution output from the greedy algorithm for an instance of Problem 1 is exactly the same near-optimal solution output from the greedy algorithm for the corresponding instance of Problem 4. □

4 Near-optimal approximation algorithms

Showing that Problem 1 is $\mathcal{NP}$-hard offers good evidence for its intractability, i.e., it is unlikely to be able to find efficient (polynomial-time) algorithms to solve it. Thus it is worth seeking for polynomial-time approximation algorithms that always output a feasible solution whose measure is not far from the global optimum.

4.1 Greedy heuristic

In [16] Tang and Liu proposed an efficient polynomial-time algorithm to solve Problem 2. Below we show that by applying the algorithm in [16] to Problem 2, the greedy algorithm proposed in [19] has the performance ratio bounded by $O(\ln n - \ln\ln n + 0.78)$, where $n = |P|$.

Let us denote the algorithm proposed in [16] to Problem 2 by `maximal-arc-intersection`. The greedy algorithm for Problem 4 is as follows.

Algorithm: `greedy-set-cover`(U, S)
1. Set $\widetilde{S} = \emptyset$ and $W = U$;
2. **while** $W \neq \emptyset$ **do**
2.1. Select a set $R \in S$ that maximizes $|R \cap W|$;
2.2. Set $\widetilde{S} = \widetilde{S} \cup \{R\}$ and $W = W \backslash R$;
3. **return** $\widetilde{S}$.

Note that Problem 1 only considers the workpiece geometry. We can further generalize it to be physically feasible. For each element $a_i \in A$, we assign a cost c_i to it. More precisely, for each workpiece setup determined by spherical polygons intersecting with a_i, to penalize physically unreasonable cutter orientation, a cost dependent on material removal rates, tolerances, workpiece stability, clamping forces, etc. is calculated and assigned to a_i. The weighted version of Problem 1 is as follows.

Problem 5 Given (1) a weight function that assigns a non-negative weight to any arc on $\mathbb{S}^2$, and (2) a set P of spherical polygons, find a set A of arcs of a given length $\theta \le 2\pi$ with a minimal sum of weight costs such that every spherical polygon intersects at least one of the arcs in A.

By Note 2, the corresponding weighted version of Problem 4 can be stated as follows.

Problem 6 Given a set system (U, S, w) with $\bigcup_{s \in S} s = U$ and weights $w : S \to \mathbb{R}_+$, find a minimum-weight set cover of (U, S).

The greedy algorithm for Problem 6 is as follows.

Algorithm: `weight-greedy-set-cover`(U, S, w)
1. Set $\widetilde{S} = \emptyset$ and $W = U$;

2. **while** $W \neq \emptyset$ **do**
2.1. Select a set $R \in S$ that minimizes $\frac{w(R)}{|R \cap W|}$;
2.2. Set $\widetilde{S} = \widetilde{S} \cup \{R\}$ and $W = W \setminus R$;
3. **return** $\widetilde{S}$.

For the set cover problem (ref. Problem 4), Johnson [12] and Lovász [13] showed that the performance ratio of the greedy algorithm `greedy-set-cover` is no worse than $H(n)$, where $H(n) = \sum_{i=1}^{n} i^{-1}$ is the nth harmonic number which has a nice property

$$\ln n < H(n) < \ln n + 1.$$

Chvátal [5] showed that the same result is held for the weighted version of the set cover problem (ref. Problem 6). For the uniform-weighted case (ref. Problem 4), a remarkably tight analysis of the `greedy-set-cover` algorithm is recently shown by Slavík [15] that its approximation radio is exactly $\ln n - \ln\ln n + \Theta(1)$, $n = |U|$. A power of the "greedy-like" algorithms for set covering problem in a priority algorithm framework introduced by Borodin et al. [3] is presented in [2]. Since the `greedy-set-cover` algorithm has the performance ratio bounded by $O(\ln n - \ln\ln n + 3 + \ln\ln 32 - \ln 32)$ (ref. Theorem 1 in [15]), by Note 2 we have the following result.

Note 3 *Given a set P of spherical polygons with cardinality n and a length $\theta \le 2\pi$, the greedy algorithm* `maximal-arc-intersection` *proposed in* [16] *outputs a set of arcs of a given length θ whose cardinality c_{greedy} satisfies*

$$\frac{c_{\text{greedy}}}{c_{\min}} \le \ln n - \ln\ln n + 3 + \ln\ln 32 - \ln 32$$
$$\approx \ln n - \ln\ln n + 0.78,$$

where $c_{\min}$ is the cardinality of the minimal set of arcs of a given length θ such that each polygon in P intersects at least one of the arcs.

4.2 Linear programming

The greedy heuristic is intuitive to an approximation solution to Problem 1. It is invited to ask "Is greedy algorithm the only approximation method for the Problem 1?" or "Is there any better approximation algorithm?"

The answer to the first question is no. Below we consider the general weighted case in Problem 5. Inspired by Note 2, we can equivalently consider the approximation solution to Problem 6, which can be reformulated as an integer programming problem:

$$\min\{w^T x | Cx \ge e\}, \tag{1}$$

where w is the weight vector for sets in S, x is a characteristic vector whose elements are only 0 or 1 and indicate whether or not the corresponding set s_i is in the output set cover, C is a characteristic matrix whose elements are also 0 or 1 and whose columns $\{c_{ij}\}_{i=1}^{m}$ are characteristic vectors of set s_j for all elements $u_i \in U$, e is a vector of unit elements.

The dual of linear programming relaxation of the optimization problem (1) is:

$$\max\{y^T e | Cy = w,\ y \ge 0\}. \tag{2}$$

Hochbaum [10] showed that the solution is the same to both problem (1) and its dual (2) if a maximal feasible solution is used. A feasible solution to the dual $\overline{y}$ is said to be maximal if there is no feasible solution y such that $y_i \ge \overline{y}_i$ and $\sum_{i=1}^{n} y_i > \sum_{i=1}^{n} \overline{y}_i$.

Let $C_{\min}$ be the unknown optimal cover for Problem 6 and C_{LP} be a maximal feasible solution to (1) and (2). By Lemma 3.2 in [11], $\frac{w(C_{\text{LP}})}{w(C_{\min})} \le \max_i\{\sum_j c_{ij}\}$, where $w(C) = \sum_{j \in C} w_j$. We have the following result.

Note 4 *Both greedy heuristic and linear programming can be used for near-optimal approximate solutions to Problems 1 and 5. The performance ratio of greedy algorithm is shown in Note 3 and the performance ratio of linear programming is bounded by* $\max_i\{\sum_j c_{ij}\}$.

Now consider the second question raised at the beginning of Sect. 4.2. If the m elements in U are uniformly distributed in the n sets in S, then $\max_i\{\sum_j c_{ij}\} \approx \frac{m}{n}$. If further m is in the linear order of n, the bound $\max_i\{\sum_j c_{ij}\}$ is tighter than $\ln n - \ln\ln n + 0.78$ for the greedy approximation.

To the best knowledge of the authors, there is no previous work that used linear programming to approximately solve the weighted maximal-arc-intersection problem of spherical polygons (ref. Problem 5).

4.3 Bound tightness of greedy heuristic

In the class of greedy approximation algorithms, the algorithm `greedy-set-cover` is one of the best known approximation algorithms to Problem 4. Raz and Safra [14] reveal that a constant c exists such that no approximation ratio of $c \ln |U|$ can be achieved, unless $\mathcal{P} = \mathcal{NP}$. Feige [6] further reveals that no approximation ratio of $c \ln |U|$ can be achieved for any $c < 1$, unless $\mathcal{NP}$ problem can be solved in $O(n^{O(\ln\ln n)})$ time.

Akin to the `greedy-set-cover` algorithm whose approximation ratio is exactly $\ln n - \ln\ln n + \Theta(1)$, $n = |U|$, an interesting question is to ask whether or not the upper bound presented in Note 3 is also tight for the greedy algorithm `maximal-arc-intersection` to Problem 1. Our answer to this question is yes, by giving the following result.

Note 5 *Problems 1 and 4 are Karp-equivalent.*

Proof Given Note 2, we can prove this result by showing that Problem 4 $\preceq$ Problem 1. Below we show that each instance of Problem 4 is reducible to a special case of Problem 1.

For Problem 1, we define a special case as follows. Let the arcs degenerate into points $\{x_i\}$ and since the polygons can be arbitrarily shaped, the polygons can be degenerated into curved paths in a graph whose vertices are $\{x_i\}$. Now given an instance of Problem 4, let each element in U correspond to a point on a sphere and let each element in S correspond to a path in a graph $G = (V, E)$, where V, E are mapped by U, S, respectively. To map this instance to a special-case instance of Problem 1, the resulting graph G needs to satisfy the following geometric constraints:

1. Each path represented by an element $s_i \in S$ must pass and only pass the vertices presented by $\{u_j : u_j \in s_i\}$;
2. Each vertex x_k represented by an element $u_k \in U$ must lie on all the paths represented by $\{s \in S : u_k \in s\}$.

To satisfy constraint 1, the edges in G may cross each other at locations that are not necessarily endpoints of edges. The so constructed G is readily seen to be a special case of Problem 1. □

5 Conclusions

In this paper, some notes on an optimal workpiece setup problem in CAD/CAM domain (Ref. Problem 1) are presented. By establishing some interesting relations between Problem 1 and two classic $\mathcal{NP}$ problems, we reveal three main results: (1) Problem 1 is $\mathcal{NP}$-hard; (2) The greedy algorithm proposed in [16, 19] to solve Problem 1 is shown to have the performance ratio bounded by $O(\ln n - \ln\ln n + 0.78)$, $n = |P|$; (3) Linear programming can also be used for a near-optimal approximation solution to Problem 1, and in some cases its performance ratio $\max_i\{\sum_j c_{ij}\}$ is better than that of greedy heuristic. We hope that these results may provide some guidelines in CAD/CAM domain, especially for the optimal workpiece setup problem.

Acknowledgements The work was supported by the Natural Science Foundation of China (Project Nos. 60736019, 60603085, 60970099) and the Tsinghua SRT Project (Project No. 091T0261).

References

1. Agarwal, P.K., Chen, D.Z., Ganjugunte, S.K., Misiolek, E., Sharir, M., Tang, K.: Stabbing convex polygons with a segment or a polygon. In: Proc. 16th Annual European Symposium on Algorithms, pp. 52–63 (2008)
2. Angelopoulos, S., Borodin, A.: The power of priority algorithms for facility location and set cover. Algorithmica **40**, 271–291 (2004)
3. Borodin, A., Nielsen, M.N., Rackoff, C.: (Incremental) priority algorithms. Algorithmica **37**, 295–326 (2003)
4. Chen, L.L., Chou, S.Y., Woo, T.: Separating and intersecting spherical polygons: Computing machinability on three-, four-, and five-axis numerically controlled machines. ACM Trans. Graph. **12**, 305–326 (1993)
5. Chvátal, V.: A greedy heuristic for the set cover problem. Math. Oper. Res. **4**, 233–235 (1979)
6. Feige, U.: A threshold of $\ln n$ for approximating set cover. J. ACM **45**, 634–652 (1998)
7. Gan, J., Woo, T., Tang, K.: Spherical maps: Their construction, properties, and approximation. J. Mech. Des. **116**, 357–363 (1994)
8. Gary, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco (1979)
9. Gupta, P., Janardan, R., Majhi, J., Woo, T.: Efficient geometric algorithms for workpiece orientation in 4- and 5-axis NC machining. Comput.-Aided Des. **28**, 577–587 (1996)
10. Hochbaum, D.S.: Approximation algorithms for the set covering and vertex cover problems. SIAM J. Comput. **11**(3), 555–556 (1982)
11. Hochbaum, D.S. (ed.): Approximation Algorithms for NP-Hard Problems. International Thomson Publishing Inc, Salt Lake City (1997)
12. Johnson, D.S.: Approximation algorithms for combinatorial problems. J. Comput. Syst. Sci. **9**, 256–278 (1974)
13. Lovász, L.: On the ratio of optimal integral and fractional covers. Discrete Math. **13**, 383–390 (1975)
14. Raz, R., Safra, S.: A sub-constant error-probability low-degree-test and a sub-constant error-probability PCP characterization of NP. In: Proceedings of the 29th Annual ACM Symposium on Theory of Computing, pp. 475–484 (1997)
15. Slavík, P.: A tight analysis of the greedy algorithm for set cover. J. Algorithms **25**, 237–254 (1997)
16. Tang, K., Liu, Y.J.: Maximum intersection of spherical polygons by an arc with applications to 4-axis machining. Comput.-Aided Des. **35**, 1269–1285 (2003)
17. Tang, K., Liu, Y.J.: A geometric method for determining intersection relations between a movable convex object and a set of planar polygons. IEEE Trans. Robot. **20**, 636–650 (2004)
18. Tang, K., Liu, Y.J.: An optimization algorithm for free-form surface partitioning based on weighted Gaussian image. Graph. Models **67**(1), 17–42 (2005)
19. Tang, K., Woo, T., Gan, J.: Maximum intersection of spherical polygons and workpiece orientation for 4- and 5-axis machining. J. Mech. Des. **114**, 477–485 (1992)
20. Tang, K., Chen, L.L., Chou, S.Y.: Optimal workpiece setups for 4-axis numerical control machining based on machinability. Comput. Ind. **37**, 27–41 (1998)

Yong-Jin Liu received his Ph.D. from the Hong Kong University of Science and Technology, Hong Kong, in 2003. He is an Associate Professor with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His research interests include computer graphics and computer-aided design.

Wen-Qi Zhang was born in 1988. He is now an undergraduate student in the Department of Computer Science and Technolgy, Tsinghua University, Beijing, China. He won the second prize in National Mathematical Olympiad and Enrolled Tsinghua University exempted from entrance examination, in 2007.

Kai Tang received his Ph.D. from the University of Michigan in 1990, in Computer Engineering. He is an Associate Professor with the Department of Mechanical Engineering, Hong Kong University of Science and Technology, Hong Kong. His research interests concentrate on designing efficient and practical algorithms for solving real world computational, geometric, and numerical problems.