Complexity 101

Ashutosh Mahajan

Department of Industrial and Systems Engineering
Lehigh University

COR@L Seminar Series, Spring 2005

Outline

COR@L
COMPUTATIONAL OPTIMIZATION RESEARCH AT LEHIGH

Why Complexity?

- Couldn't think of any other topic.
- Fairly interesting.

I conjecture that there is no good algorithm for the traveling salesman problem. My reasons are the same for any mathematical conjecture: (1) It is a legitimate mathematical possibility. (2) I do not know. Jack Edmonds, 1966.

- Could be useful.

Why Complexity?

- Couldn't think of any other topic.
- Fairly interesting.

I conjecture that there is no good algorithm for the traveling salesman problem. My reasons are the same for any mathematical conjecture: (1) It is a legitimate mathematical possibility. (2) I do not know. Jack Edmonds, 1966.

- Could be useful.

COR@L
COMPUTATIONAL OPTIMIZATION RESEARCH AT LEHIGH

Why Complexity?

- Couldn't think of any other topic.
- Fairly interesting.

I conjecture that there is no good algorithm for the traveling salesman problem. My reasons are the same for any mathematical conjecture: (1) It is a legitimate mathematical possibility. (2) I do not know. Jack Edmonds, 1966.

- Could be useful.

Why Complexity?

- Couldn't think of any other topic.
- Fairly interesting.

I conjecture that there is no good algorithm for the traveling salesman problem. My reasons are the same for any mathematical conjecture: (1) It is a legitimate mathematical possibility. (2) I do not know. Jack Edmonds, 1966.

- Could be useful.

Why Complexity?

- Couldn't think of any other topic.
- Fairly interesting.

I conjecture that there is no good algorithm for the traveling salesman problem. My reasons are the same for any mathematical conjecture: (1) It is a legitimate mathematical possibility. (2) I do not know. Jack Edmonds, 1966.

- Could be useful.

Basics

- Formal Vs Informal
- Simple Turing Machine $M = (K, \Sigma, \delta, s)$.
- Language $L \subset (\Sigma - \{\sqcup\})^*$

Basics

- Formal Vs Informal
- Simple Turing Machine $M = (K, \Sigma, \delta, s)$.
- Language $L \subset (\Sigma - \{\sqcup\})^*$

Basics

- Formal Vs Informal
- Simple Turing Machine $M = (K, \Sigma, \delta, s)$.
- Language $L \subset (\Sigma - \{\sqcup\})^*$

Languages

- Language is informally an input given to M
- Recursive Language
- Recursively Enumerable Language
- Recursive $\subset$ Recursively Enumerable

Languages

- Language is informally an input given to M
- Recursive Language
- Recursively Enumerable Language
- Recursive $\subset$ Recursively Enumerable

Languages

- Language is informally an input given to M
- Recursive Language
- Recursively Enumerable Language
- Recursive $\subset$ Recursively Enumerable

Languages

- Language is informally an input given to M
- Recursive Language
- Recursively Enumerable Language
- Recursive $\subset$ Recursively Enumerable

Pallindromes

- *νιψον ανομηματα μη μοναν οψιν*
- *Wash away my sins not just my face.*
- Byzantine church of Aghia Sophia, Constantinople (Istanbul)
- **TIME?**
- **TIME**$(\frac{1}{2}(n+1)(n+2))$

Pallindromes

- $\nu\iota\psi o\nu\ \alpha\nu o\mu\eta\mu\alpha\tau\alpha\ \mu\eta\ \mu o\nu\alpha\nu\ o\psi\iota\nu$
- *Wash away my sins not just my face.*
- Byzantine church of Aghia Sophia, Constantinople (Istanbul)
- **TIME?**
- **TIME**$(\frac{1}{2}(n+1)(n+2))$

COR@L
COMPUTATIONAL OPTIMIZATION RESEARCH AT LEHIGH

Pallindromes

- $\nu\iota\psi o\nu\ \alpha\nu o\mu\eta\mu\alpha\tau\alpha\ \mu\eta\ \mu o\nu\alpha\nu\ o\psi\iota\nu$
- *Wash away my sins not just my face.*
- Byzantine church of Aghia Sophia, Constantinople (Istanbul)
- **TIME?**
- **TIME**$(\frac{1}{2}(n+1)(n+2))$

Pallindromes

- $\nu\iota\psi o\nu\ \alpha\nu o\mu\eta\mu\alpha\tau\alpha\ \mu\eta\ \mu o\nu\alpha\nu\ o\psi\iota\nu$
- *Wash away my sins not just my face.*
- Byzantine church of Aghia Sophia, Constantinople (Istanbul)
- **TIME?**
- **TIME**$(\frac{1}{2}(n+1)(n+2))$

Pallindromes

- *νιψον ανομηματα μη μοναν οψιν*
- *Wash away my sins not just my face.*
- Byzantine church of Aghia Sophia, Constantinople (Istanbul)
- **TIME?**
- **TIME**$(\frac{1}{2}(n+1)(n+2))$

Pallindromes

- $\nu\iota\psi o\nu\ \alpha\nu o\mu\eta\mu\alpha\tau\alpha\ \mu\eta\ \mu o\nu\alpha\nu\ o\psi\iota\nu$
- *Wash away my sins not just my face.*
- Byzantine church of Aghia Sophia, Constantinople (Istanbul)
- **TIME?**
- **TIME**$(\frac{1}{2}(n+1)(n+2))$

k-string Turing Machines

- Let M have 2 reading heads
- Pallindromes can be detected in **TIME**$(3n+3)$.

k-string Turing Machines

- Let M have 2 reading heads
- Pallindromes can be detected in **TIME**$(3n + 3)$.

Big Results

- Given any k-string Turing Machine M operating within time $f(n)$, we can construct a Turing Machine M' operating within time $\mathcal{O}(f(n)^2)$ and such that, for any input $x, M(x) = M'(x)$.
- Similar result for RAMs. ($\mathcal{O}(f(n)^3)$).
- Let $L \in$**TIME**$(f(n))$. Then, for any $\epsilon > 0, L \in$ **TIME**$(f'(n))$, where $f'(n) = \epsilon f(n) + n + 2$.
- Thus any conceivable machine can only be polynomially better than a Turing Machine.

Big Results

- Given any k-string Turing Machine M operating within time $f(n)$, we can construct a Turing Machine M' operating within time $\mathcal{O}(f(n)^2)$ and such that, for any input $x, M(x) = M'(x)$.
- Similar result for RAMs. ($\mathcal{O}(f(n)^3)$).
- Let $L \in$**TIME**$(f(n))$. Then, for any $\epsilon > 0, L \in$ **TIME**$(f'(n))$, where $f'(n) = \epsilon f(n) + n + 2$.
- Thus any conceivable machine can only be polynomially better than a Turing Machine.

Big Results

- Given any k-string Turing Machine M operating within time $f(n)$, we can construct a Turing Machine M' operating within time $\mathcal{O}(f(n)^2)$ and such that, for any input $x, M(x) = M'(x)$.
- Similar result for RAMs. ($\mathcal{O}(f(n)^3)$).
- Let $L \in$**TIME**$(f(n))$. Then, for any $\epsilon > 0, L \in$ **TIME**$(f'(n))$, where $f'(n) = \epsilon f(n) + n + 2$.
- Thus any conceivable machine can only be polynomially better than a Turing Machine.

Big Results

- Given any k-string Turing Machine M operating within time $f(n)$, we can construct a Turing Machine M' operating within time $\mathcal{O}(f(n)^2)$ and such that, for any input $x, M(x) = M'(x)$.
- Similar result for RAMs. $(\mathcal{O}(f(n)^3))$.
- Let $L \in$**TIME**$(f(n))$. Then, for any $\epsilon > 0, L \in$ **TIME**$(f'(n))$, where $f'(n) = \epsilon f(n) + n + 2$.
- Thus any conceivable machine can only be polynomially better than a Turing Machine.

Nondeterministic Turing Machines

- $N = (K, \Sigma, \Delta, s)$
- Δ is no longer a function.
- N decides L if for any $x \in \Sigma^*$, the following is true:
- $x \in L$ iff $(s, \triangleright, x) \rightarrow (\text{yes}, w, u)$ for some strings w, u.
- N gives a yes because of some sequence of *non-deterministic* choices.
- Unreasonable model of computation?
- $\mathbf{P} = \bigcup \mathbf{TIME}(n^k)$
- $\mathbf{NP} = \bigcup \mathbf{NTIME}(n^k)$

Nondeterministic Turing Machines

- $N = (K, \Sigma, \Delta, s)$
- Δ is no longer a function.
- N decides L if for any $x \in \Sigma^*$, the following is true:
- $x \in L$ iff $(s, \triangleright, x) \rightarrow (\text{yes}, w, u)$ for some strings w, u.
- N gives a yes because of some sequence of *non-deterministic* choices.
- Unreasonable model of computation?
- **P** $= \bigcup$**TIME**(n^k)
- **NP** $= \bigcup$**NTIME**(n^k)

Nondeterministic Turing Machines

- $N = (K, \Sigma, \Delta, s)$
- Δ is no longer a function.
- N decides L if for any $x \in \Sigma^*$, the following is true:
- $x \in L$ iff $(s, \triangleright, x) \rightarrow$ (yes, w, u) for some strings w, u.
- N gives a yes because of some sequence of *non-deterministic* choices.
- Unreasonable model of computation?
- $\mathbf{P} = \bigcup \mathbf{TIME}(n^k)$
- $\mathbf{NP} = \bigcup \mathbf{NTIME}(n^k)$

Nondeterministic Turing Machines

- $N = (K, \Sigma, \Delta, s)$
- Δ is no longer a function.
- N decides L if for any $x \in \Sigma^*$, the following is true:
- $x \in L$ iff $(s, \triangleright, x) \rightarrow$ (yes, w, u) for some strings w, u.
- N gives a yes because of some sequence of *non-deterministic* choices.
- Unreasonable model of computation?
- **P** = $\bigcup$**TIME**(n^k)
- **NP** = $\bigcup$**NTIME**(n^k)

Nondeterministic Turing Machines

- $N = (K, \Sigma, \Delta, s)$
- Δ is no longer a function.
- N decides L if for any $x \in \Sigma^*$, the following is true:
- $x \in L$ iff $(s, \triangleright, x) \rightarrow (\text{yes}, w, u)$ for some strings w, u.
- N gives a yes because of some sequence of *non-deterministic* choices.
- Unreasonable model of computation?
- $\mathbf{P} = \bigcup \mathbf{TIME}(n^k)$
- $\mathbf{NP} = \bigcup \mathbf{NTIME}(n^k)$

Nondeterministic Turing Machines

- $N = (K, \Sigma, \Delta, s)$
- Δ is no longer a function.
- N decides L if for any $x \in \Sigma^*$, the following is true:
- $x \in L$ iff $(s, \triangleright, x) \rightarrow$ (yes, w, u) for some strings w, u.
- N gives a yes because of some sequence of *non-deterministic* choices.
- Unreasonable model of computation?
- **P** $= \bigcup$**TIME**(n^k)
- **NP** $= \bigcup$**NTIME**(n^k)

Nondeterministic Turing Machines

- $N = (K, \Sigma, \Delta, s)$
- Δ is no longer a function.
- N decides L if for any $x \in \Sigma^*$, the following is true:
- $x \in L$ iff $(s, \triangleright, x) \to (\text{yes}, w, u)$ for some strings w, u.
- N gives a yes because of some sequence of *non-deterministic* choices.
- Unreasonable model of computation?
- $\mathbf{P} = \bigcup \mathbf{TIME}(n^k)$
- $\mathbf{NP} = \bigcup \mathbf{NTIME}(n^k)$

Nondeterministic Turing Machines

- $N = (K, \Sigma, \Delta, s)$
- Δ is no longer a function.
- N decides L if for any $x \in \Sigma^*$, the following is true:
- $x \in L$ iff $(s, \triangleright, x) \rightarrow$ (yes, w, u) for some strings w, u.
- N gives a yes because of some sequence of *non-deterministic* choices.
- Unreasonable model of computation?
- **P** = $\bigcup$**TIME**(n^k)
- **NP** = $\bigcup$**NTIME**(n^k)

Universal Turing Machine

- $U(M; x) = M(x)$
- M halts if and only if U halts.

Universal Turing Machine

- $U(M;x) = M(x)$
- M halts if and only if U halts.

Our first problem!

- Given the description of a machine M and its input x, will M halt on x?
- The famous Halting Problem (H).
- H is recursively enumerable
- H is NOT recursive
- $D(M)$: if $M_H(M; M)$ =yes, then LOOP forever, else YES
- What is $D(D)$?
- Any non-trivial property of Turing Machines is undecidable !!
- Incompleteness Theorem?

Our first problem!

- Given the description of a machine M and its input x, will M halt on x?
- The famous Halting Problem (H).
- H is recursively enumerable
- H is NOT recursive
- $D(M)$: if $M_H(M; M)$ =yes, then LOOP forever, else YES
- What is $D(D)$?
- Any non-trivial property of Turing Machines is undecidable !!
- Incompleteness Theorem?

Our first problem!

- Given the description of a machine M and its input x, will M halt on x?
- The famous Halting Problem (H).
- H is recursively enumerable
- H is NOT recursive
- $D(M)$: if $M_H(M;M)$ =yes, then LOOP forever, else YES
- What is $D(D)$?
- Any non-trivial property of Turing Machines is undecidable !!
- Incompleteness Theorem?

Our first problem!

- Given the description of a machine M and its input x, will M halt on x?
- The famous Halting Problem (H).
- H is recursively enumerable
- H is NOT recursive
- $D(M)$: if $M_H(M; M)$ =yes, then LOOP forever, else YES
- What is $D(D)$?
- Any non-trivial property of Turing Machines is undecidable !!
- Incompleteness Theorem?

Our first problem!

- Given the description of a machine M and its input x, will M halt on x?
- The famous Halting Problem (H).
- H is recursively enumerable
- H is NOT recursive
- $D(M)$: if $M_H(M; M)$ =yes, then LOOP forever, else YES
- What is $D(D)$?
- Any non-trivial property of Turing Machines is undecidable !!
- Incompleteness Theorem?

Our first problem!

- Given the description of a machine M and its input x, will M halt on x?
- The famous Halting Problem (H).
- H is recursively enumerable
- H is NOT recursive
- $D(M)$: if $M_H(M;M)$ =yes, then LOOP forever, else YES
- What is $D(D)$?
- Any non-trivial property of Turing Machines is undecidable !!
- Incompleteness Theorem?

Our first problem!

- Given the description of a machine M and its input x, will M halt on x?
- The famous Halting Problem (H).
- H is recursively enumerable
- H is NOT recursive
- $D(M)$: if $M_H(M; M) =$yes, then LOOP forever, else YES
- What is $D(D)$?
- Any non-trivial property of Turing Machines is undecidable !!
- Incompleteness Theorem?

Our first problem!

- Given the description of a machine M and its input x, will M halt on x?
- The famous Halting Problem (H).
- H is recursively enumerable
- H is NOT recursive
- $D(M)$: if $M_H(M; M)$ =yes, then LOOP forever, else YES
- What is $D(D)$?
- Any non-trivial property of Turing Machines is undecidable !!
- Incompleteness Theorem?

R, RE, coRE

- If L is recursive, then so is $\bar{L}$.
- L is recursive iff L and $\bar{L}$ are recursively enumerable
- Pictorially ...
- Severe Undecidability: Where shall we go for dinner tonight?

R, RE, $coRE$

- If L is recursive, then so is $\bar{L}$.
- L is recursive iff L and $\bar{L}$ are recursively enumerable
- Pictorially ...
- Severe Undecidability: Where shall we go for dinner tonight?

R, RE, $coRE$

- If L is recursive, then so is $\bar{L}$.
- L is recursive iff L and $\bar{L}$ are recursively enumerable
- Pictorially ...
- Severe Undecidability: Where shall we go for dinner tonight?

R, RE, $coRE$

- If L is recursive, then so is $\bar{L}$.
- L is recursive iff L and $\bar{L}$ are recursively enumerable
- Pictorially ...
- Severe Undecidability: Where shall we go for dinner tonight?