ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA

LÂM HOÀNG VŨ

DỰ BÁO CHUỖI THỜI GIAN SỬ DỤNG MÔ HÌNH ARIMA VÀ GIẢI THUẬT DI TRUYỀN

Chuyên Ngành: Khoa Học Máy Tính
Mã số: 60.48.01

LUẬN VĂN THẠC SĨ

TP. HỒ CHÍ MINH, tháng 07 năm 2012

ĐẠI HỌC QUỐC GIA TP. HCM
TRƯỜNG ĐẠI HỌC BÁCH KHOA

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc Lập - Tự Do - Hạnh Phúc

---------- **---oOo---**

Tp. HCM, ngày. .. . tháng. .. . năm .2012.

NHIỆM VỤ LUẬN VĂN THẠC SĨ

Họ và tên học viên: Lâm Hoàng Vũ…….....Giới tính: Nam □/ Nữ □

Ngày, tháng, năm sinh: 14/10/1981….....Nơi sinh: Quảng Ngãi

Chuyên ngành: Khoa học Máy tính……...

Khoá: 2008……

1- TÊN ĐỀ TÀI:

DỰ BÁO CHUỖI THỜI GIAN SỬ DỤNG MÔ HÌNH ARIMA VÀ GIẢI THUẬT DI TRUYỀN

.....

2- NHIỆM VỤ LUẬN VĂN:

.....

.....

.....

.....

3- NGÀY GIAO NHIỆM VỤ:

.....

4- NGÀY HOÀN THÀNH NHIỆM VỤ:

.....

5- HỌ VÀ TÊN CÁN BỘ HƯỚNG DẪN: PGS TS. Dương Tuấn Anh…….…….

Nội dung và đề cương Luận văn thạc sĩ đã được Hội Đồng Chuyên Ngành thông qua.

CÁN BỘ HƯỚNG DẪN **CHỦ NHIỆM BỘ MÔN**

(Họ tên và chữ ký) **QUẢN LÝ CHUYÊN NGÀNH**

(Họ tên và chữ ký)

PGS TS. Dương Tuấn Anh TS. Đinh Đức Anh Vũ

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI
TRƯỜNG ĐẠI HỌC BÁCH KHOA
ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH

Cán bộ hướng dẫn khoa học: PGS TS. Dương Tuấn Anh.....

Cán bộ chấm nhận xét 1: .....

.....

.....

.....

.....

Cán bộ chấm nhận xét 2: .....

.....

.....

.....

.....

Luận văn thạc sĩ được bảo vệ tại

.....

.....

HỘI ĐỒNG CHẤM BẢO VỆ LUẬN VĂN THẠC SĨ

TRƯỜNG ĐẠI HỌC BÁCH KHOA, Ngày . . . . Tháng . . . . Năm. 2012

LỜI CAM ĐOAN

Tôi cam đoan rằng, ngoại trừ các kết quả tham khảo từ các công trình khác như đã ghi rõ trong luận văn, các công việc trình bày trong luận văn này là do chính tôi thực hiện và chưa có phần nội dung nào của luận văn này được nộp để lấy một bằng cấp ở trường này hoặc trường khác.

Ngày 01 tháng 07 năm 2012

Lâm Hoàng Vũ

LỜI CÁM ƠN

Tôi xin bày tỏ lòng biết ơn chân thành nhất đến PGS.TS. Dương Tuấn Anh, Thầy đã tận tâm chỉ dẫn, truyền đạt những kiến thức và kinh nghiệm quý báu cho tôi từ những ngày đầu cũng như những ngày cuối trong suốt quá trình thực hiện luận văn này.

Tôi cũng xin được gửi lời cám ơn đến các quí Thầy Cô giáo tham gia giảng dạy chương trình cao học ở khoa Khoa Học và Kỹ Thuật Máy Tính, trường Đại Học Bách Khoa TP. Hồ Chí Minh đã trang bị cho tôi những kiến thức nền tảng quan trọng trong suốt quá trình tôi theo học.

Và cuối cùng, tôi xin được gửi lời cám ơn đến gia đình và bạn bè, những người đã động viên tôi trong suốt thời gian vừa qua.

TÓM TẮT LUẬN VĂN

Các nghiên cứu về dữ liệu chuỗi thời gian đem lại những ứng dụng thực tế quan trọng trong các lĩnh vực như thống kê, xử lý tín hiệu số, toán tài chính, … Một trong số đó là bài dự báo chuỗi thời gian **(**hay dự báo các giá trị tương lai của chuỗi thời gian từ các giá trị trong quá khứ) từ việc xây dựng các mô hình dự báo thích hợp.

Đã có nhiều nghiên cứu tập trung vào bài toán dự báo chuỗi thời gian, một trong số đó là sử dụng mô hình ARIMA, mô hình ARMA, trong đó việc lựa chọn mô hình dựa theo phương pháp của Box-Jenkins và việc ước lượng các hệ số của mô hình dựa trên các phương pháp toán học thuần túy rất phức tạp. Hơn nữa, kết quả của phương pháp Box-Jenkins phụ thuộc rất nhiều vào năng lực chuyên môn của người làm dự báo. Để giải quyết vấn đề này, có nhiều phương pháp meta-heuristic sử dụng giải thuật di truyền được đề xuất để việc lựa chọn mô hình (thể hiện qua bậc và các biến thời gian trễ có mặt trong mô hình) và tính toán các hệ số của mô hình một cách tự động. Tuy vậy, việc sinh ra các mô hình trong quá trình tìm kiếm lời giải của các phương pháp meta-heuristic được thực hiện mang tính chất ngẫu nhiên (bởi bản chất của các giải thuật di truyền, giải thuật mô phỏng luyện kim) và các phương pháp meta-heuristic này thường chạy rất chậm để cho ra lời giải tốt.

Từ những vấn đề nêu trên, trong đề tài này, cũng với mục tiêu đưa ra một phương pháp để tự động xác định bậc và ước lượng các hệ số của mô hình ARMA, chúng tôi đề xuất một phương pháp mở rộng không gian tìm kiếm các lời giải của mô hình ARMA dựa trên giải thuật tìm kiếm Tabu trong việc xác định bậc và sử dụng giải thuật di truyền để ước lượng các hệ số của mô hình ARMA. Kết quả thực nghiệm cho thấy phương pháp mới này đem lại kết quả tốt hơn đối với hầu hết các chuỗi dữ liệu được kiểm tra so với các phương pháp meta-heuristic khác và thời gian chạy dừng ở mức có thể chấp nhận được.

MỤC LỤC

DANH MỤC HÌNH

DANH MỤC BẢNG

DANH MỤC TỪ VIẾT TẮT

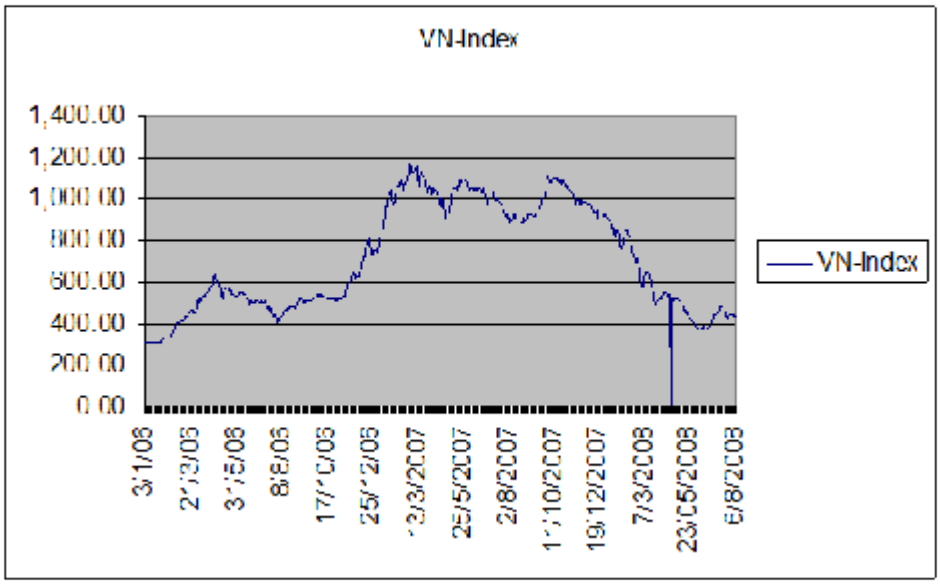
ACF	Hàm tự tương quan (Autocorrelation Function)
ACVF	Hàm tự hiệp phương sai (Autocovariance Function)
ANN	Mạng nơ-ron nhân tạo (Artificial Neural Network)
AR	Tự hồi qui (Autoregression)
ARIMA	Tự hồi qui kết hợp trung bình di động (Autoregression Integrated Moving Average)
ARMA	Tự hồi qui – Trung bình di động (Autoregression Moving Average)
EWMA	Trung bình di động có trọng số theo mũ (Exponentially Weighted Moving Average)
GA	Giải thuật di truyền (Genetic Algorithm)
HMM	Mô hình Markov ẩn (Hidden Markov Model)
MA	Trung bình di động (Moving Average)
NST	Nhiễm sắc thể (Chromosome)
PACF	Hàm tự tương quan riêng phần (Partial Autocorrelation Function)

Chương 1. GIỚI THIỆU

1.1 Dữ liệu chuỗi thời gian

1.1.1 Định nghĩa

Chuỗi thời gian là một tập hợp dữ liệu các quan sát đo được một cách tuần tự theo thời gian. Các quan sát này có thể đo được một cách liên tục theo thời gian hoặc là có thể được lấy theo một tập rời rạc các thời điểm khác nhau.



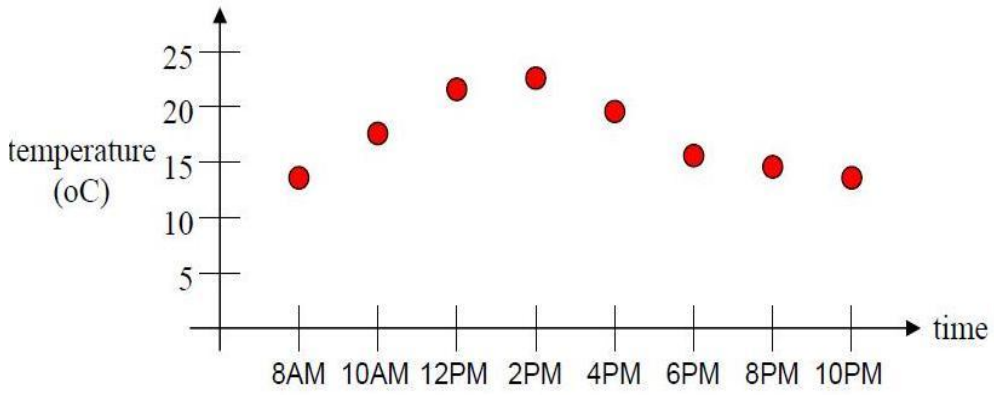
Hình 1.1: Đường biểu diễn dữ liệu chuỗi thời gian cho chỉ số VN-Index từ ngày 3/1/2006 đến ngày 6/8/2008

Theo qui ước về cách tạo dữ liệu chuỗi dữ liệu thời gian như trên, ta lần lượt gọi hai kiểu chuỗi này là *chuỗi thời gian liên tục* và *chuỗi thời gian rời rạc* ngay cả khi biến đo được là biến rời rạc trong trường hợp chuỗi thời gian liên tục và lại là biến liên tục trong trường hợp chuỗi thời gian rời rạc.

Giá trị của chuỗi tuần tự theo thời gian của đại lượng X được ký hiệu $X = < x_1, x_2, \dots, x_t, \dots, x_n >$ với x_t là giá trị quan sát của X ở thời điểm t và n được gọi là chiều dài của chuỗi quan sát. Sự chuyển tiếp từ thời gian này sang thời gian khác được gọi là bước.

Các giá trị quan sát có thể được ghi nhận ở những khoảng thời gian không bằng nhau. Tuy nhiên ta chỉ quan tâm tới chuỗi thời gian là chuỗi mà các giá trị là rời rạc và được ghi nhận ở những khoảng thời gian cố định bằng nhau và trong hầu hết các ứng dụng thực tế, dữ liệu được đo cách nhau trong một khoảng thời gian cố định để đơn giản hóa quá trình lưu trữ cũng như độ phức tạp của dữ liệu.

Ví dụ ta có chuỗi thời gian theo dõi quá trình đo nhiệt độ S=<14.3, 18.2, 22.0, 22, 4, 19.5, 17.1, 15.8, 15.1> (xem hình 1.2).



Hình 1.2: Minh họa về dữ liệu chuỗi thời gian theo dõi quá trình đo nhiệt độ

1.1.2 Các thành phần của chuỗi thời gian

Dữ liệu của chuỗi thời gian thường bao gồm 4 thành phần:

- Thành phần xu hướng dài hạn (***T***): Thành phần này dùng để chỉ xu hướng tăng giảm của đại lượng X trong khoảng thời gian dài.

- Thành phần mùa (***S***): Thành phần này chỉ sự thay đổi của đại lượng *X* theo các mùa trong năm.
- Thành phần chu kỳ (***C***): Thành phần này chỉ sự thay đổi của đại lượng *X* theo chu kỳ. Sự khác biệt của thành phần này so với thành phần mùa là chu kỳ của nó dài hơn một năm.
- Thành phần bất thường (***I***): Thành phần này dùng để chỉ những sự thay đổi bất thường của các giá trị trong chuỗi tuần tự theo thời gian. Sự thay đổi này không thể dự đoán bằng các số liệu kinh nghiệm trong quá khứ, về mặt bản chất thành phần này không có tính chu kỳ.

1.1.3 Ứng dụng của phân tích dữ liệu chuỗi thời gian

Chuỗi thời gian được sử dụng để thu thập các dữ liệu quan sát trong rất nhiều lĩnh vực như thống kê, xử lý tín hiệu số, toán tài chính… trước khi thực hiện các phân tích thích hợp tùy vào ứng dụng của mỗi lĩnh vực cụ thể. Phân tích chuỗi thời gian nhằm mục đích rút trích được các thống kê có ý nghĩa, giải quyết vấn đề nhận diện những đặc trưng cơ bản của chuỗi thời gian cũng như là khai phá cấu trúc nội tại của chuỗi thời gian từ dữ liệu quan sát được. Những mục tiêu chính của việc phân tích chuỗi thời gian là:

- **xây dựng các mô hình input-output** mô tả các hàm biến đổi tương đương theo chuỗi thời gian
- **dự báo chuỗi thời gian** hay dự báo các giá trị tương lai của chuỗi thời gian từ các giá trị trong quá khứ từ việc sử dụng các mô hình đã được xây dựng
- **thiết kế các hệ thống điều khiển:** kết quả dự báo tốt cho phép người phân tích thực hiện điều khiển một quá trình cụ thể nào đó, nó có thể là một qui trình công nghiệp, kinh tế, …

Ngoài các mục tiêu kể trên, các lớp bài toán liên quan đến dữ liệu chuỗi thời gian là khá rộng, chẳng hạn như các bài toán tìm kiếm tương tự (similarity search), gom cụm dữ liệu (clustering), phân loại dữ liệu (classification), tìm qui luật của dữ liệu (rule discovery), phát hiện điểm bất thường (novelty dectection), tìm mẫu lặp (finding motif). Áp dụng các bài toán nêu trên có thể giải giải quyết các ứng dụng thực tế sau đây:

- Ứng dụng nhận dạng chữ viết tay: chữ viết được biểu diễn dưới dạng dữ liệu chuỗi thời gian. Việc so trùng dữ liệu của hai chuỗi thời gian sẽ cho ta biết chúng có tương tự nhau không, từ đó suy ra hai dạng chữ viết có phải của cùng một người hay không.
- Xác định những mã chứng khoán có giá biến động theo cùng một kiểu giống nhau.

1.1.4 Một số vấn đề thường gặp khi nghiên cứu chuỗi thời gian

- **Khối lượng dữ liệu:** một trong những đặc trưng của chuỗi thời gian là dữ liệu rất lớn, đây là một trong những vấn đề thách thức trong quá trình phân tích, tính toán và xử lý dữ liệu chuỗi thời gian để tạo ra kết quả chính xác trong thời gian hợp lý.
- **Phụ thuộc yếu tố chủ quan:** trong thực tế, các kết quả dữ liệu chuỗi thời gian thu được chịu ảnh hưởng yếu tố chủ quan của người đo dữ liệu, điều kiện và các công cụ đo…
- **Dữ liệu không đồng nhất:** quá trình thu thập dữ liệu chuỗi thời gian được đo trên những định dạng khác nhau, số lượng và tần số lấy mẫu không đồng nhất cũng ảnh hưởng đến tính toàn vẹn của dữ liệu. Thêm vào đó quá trình đo đạc không chính xác do nhiễu, thiếu một vài giá trị hay dữ liệu không sạch.

Phần tiếp theo sẽ trình bày chi tiết về một trong những bài toán lớn của dữ liệu chuỗi thời gian là bài toán dự báo.

1.2 Bài toán dự báo chuỗi thời gian

Nghiên cứu khoa học về các đối tượng nào đó (chẳng hạn như các hệ trong vật lý hoặc một vấn đề nào đó trong kinh tế) thường dựa vào các chuỗi thời gian tạo ra từ dữ liệu các mẫu quan sát được theo thời gian, dữ liệu này chính là cơ sở để hiểu được đặc tính cũng như là dự đoán các hành vi tương lai của đối tượng đó. Nếu ta xác định được những phương trình cơ sở thì các đối tượng nghiên cứu này có thể phân tích được và qua đó xác định được các đặc tính của chúng. Tuy nhiên, trong thực tế, ta thường không biết được những phương trình cơ sở của đối tượng nghiên cứu. Trong trường hợp này, những qui tắc quan sát được trong quá khứ sẽ được sử dụng như là những chỉ dẫn để hiểu được đối tượng nghiên cứu và dự đoán hành vi tương lai.

Định nghĩa bài toán: Cho một dãy các dữ liệu quan sát được theo thời gian, một hệ thống dự báo sẽ thực hiện việc ước lượng các giá trị quan sát trong vài chu kỳ thời kế tiếp. Ta định nghĩa bài toán một cách chi tiết như sau:

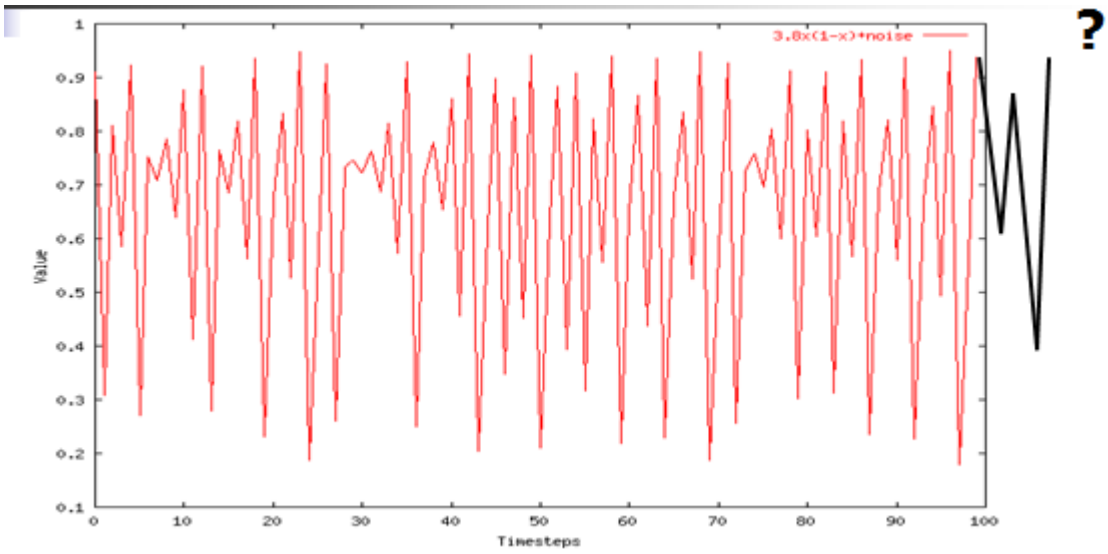
Dự báo 1-bước: Cho trước dãy $x_1, x_2, \dots, x_t$, dự đoán giá trị của x_{t+1}.

Bài toán này được tổng quát hóa thành bài sau:

Dự báo n-bước: Tập huấn luyện (còn gọi là tập dữ liệu quan sát được trong quá khứ) là một tập hợp các chuỗi thời gian tạo ra từ cùng một đối tượng nghiên cứu trên các chu kỳ thời gian khác nhau.

$$TS = \{X_1, X_2, \dots, X_N\}$$

Với $X_i = x_{t_i}, x_{t_i+1}, \dots, x_{t_i+(l_i-1)}$, trong đó x_t là giá trị của chuỗi thời gian tại thời điểm t và l_i là độ dài của dãy X_i. Hệ thống dự báo sẽ được cung cấp tương ứng với tập TS dãy kết quả truy vấn $Y = y_1, y_2, \dots, y_l$ và ta sẽ cần tìm các giá trị $y_{l+1}, y_{l+2}, \dots$



Hình 1.3: Đồ thị chuỗi thời gian và các giá trị dự báo

Phân tích chuỗi thời gian cho mục đích dự báo là một mảng nghiên cứu lớn với các ứng dụng rộng lớn đa dạng. Những liệt kê sau đây cho thấy phần nào những lĩnh vực mà ứng dụng của dự báo chuỗi thời gian đã được chứng thực [1].

- Vật lý: đo độ dao động của tia laser.
- Sinh học: dữ liệu sinh lý học của các bệnh nhân mắc chứng ngưng thở lúc ngủ như nhịp tim, độ tập trung oxy trong máu, trạng thái điện não đồ.
- Kinh tế: dữ liệu về tỷ giá trao đổi ngoại tệ, chỉ số chứng khoán hàng ngày.
- Thiên văn học: mật độ biến đổi của các sao lùn trắng, tiên đoán hoạt động của năng lượng mặt trời.
- Địa vật lý: các phép đo dữ liệu bão từ tính.

1.3 Động cơ và mục tiêu nghiên cứu

Mô hình ARIMA (*tự hồi qui kết hợp trung bình di động*) là một công cụ mạnh mẽ để áp dụng vào việc phân tích và dự báo các chuỗi thời gian. Tuy nhiên câu hỏi đặt ra là khi nào thì cần đến mô hình ARIMA (nghĩa là làm thế nào biết được chuỗi thời gian

quan sát là phù hợp với mô hình ARIMA) và lựa chọn mô hình ARIMA cụ thể (tức là xác định bậc của mô hình) như thế nào để sử dụng?

Cách tiếp cận phổ biến được biết đến nhiều nhất cho vấn đề lựa chọn mô hình là phương pháp Box-Jenkins [2], và mô hình ARIMA là một trong số các mô hình của phương pháp này cùng với các mô hình khác như: AR, MA, ARMA. Phương pháp Box-Jenkins bao gồm các bước: phân tích nhận dạng mẫu quan sát (sử dụng các số liệu quan sát được để phân tích và tìm ra mô hình thích hợp); ước lượng các tham số của mô hình và kiểm tra chẩn đoán sự phù hợp của mô hình.

Kết quả của phương pháp Box-Jenkins tùy thuộc rất lớn vào năng lực và kinh nghiệm của người phân tích. Đặc biệt, ở bước phân tích nhận dạng mẫu, giá trị tương quan giữa các mẫu sẽ xác định được giá trị tối ưu cho bậc của các thành phần AR và MA trong mô hình ARIMA. Thế nhưng ta thường thấy rằng các mô hình khác nhau có thể có các giá trị tương quan tương tự nhau và như vậy việc lựa chọn mô hình trong số các mô hình ứng viên có tính tùy tiện.

Mục tiêu nghiên cứu của đề tài này là đưa ra một phương pháp tính toán tự động chọn ra mô hình phù hợp nhất trong lớp các mô hình ARIMA dựa vào giải thuật di truyền. Giải thuật di truyền lấy ý tưởng từ quá trình chọn lọc tự nhiên trong sinh học là một công cụ mạnh mẽ để giải quyết các bài toán tìm kiếm và tối ưu hóa. Đối với mô hình ARIMA, ta sẽ xây dựng một giải thuật di truyền phù hợp để sử dụng được vào cả hai mục đích:

- xác định bậc phù hợp cho các thành phần AR và MA có trong mô hình ARIMA
- xác định các hệ số của mô hình

1.4 Tóm lược các kết quả đạt được

- Xây dựng mô hình GA-ARMA sử dụng giải thuật di truyền để ước lượng các tham số của mô hình ARMA, trong đó giải thuật di truyền có sử dụng đến các biến thể mới của các phép toán lai ghép, đột biến cho trường hợp biểu diễn số thực.
- Chúng tôi đã đề xuất một phương pháp mở rộng không gian tìm kiếm các mô hình ARMA khác với các phương pháp được thực hiện bởi Cortez và các cộng sự trong [5] bằng cách xây dựng nên một biến thể của giải thuật tìm kiếm Tabu chuẩn.

1.5 Cấu trúc của luận văn

Luận văn tốt nghiệp gồm các phần như sau:

- Chương 1 như vừa trình bày giới thiệu về bài toán dự báo chuỗi thời gian, động cơ nghiên cứu của bài toán này mục tiêu cần nghiên cần nghiên cứu để giải quyết bài toán này.
- Chương 2 trình bày tổng quan về phương pháp và các mô hình dự báo chuỗi thời gian. Đặc biệt trong chương này, chúng tôi cũng điểm qua các công trình nhận dạng và ước lượng tham số của mô hình ARMA sử dụng các phương pháp meta-heuristic khác. Những công trình này góp phần làm nền tảng để chúng tôi đưa ra một phương pháp khác để nhận dạng và ước lượng tham số mô hình ARMA trong luận văn này.
- Chương 3 là cơ sở lý thuyết để hình thành nên cách tiếp cận và giải quyết vấn đề của luận văn sắp tới. Ở chương này giới thiệu về chuỗi thời gian tĩnh (là các chuỗi dữ liệu thích hợp cho mô hình ARMA), mô hình ARIMA và các thành phần của nó. Trong chương này chúng tôi cũng đi sâu vào việc tìm hiểu phương pháp sử dụng giải thuật di truyền để xác định bậc của mô hình ARMA, phương

pháp sử dụng giải thuật di truyền để ước lượng tham số của mô hình của các nhóm tác giả khác nhau mà tiêu biểu là phương pháp siêu tiến hóa của Cortez (2001) [5].

- Chương 4 trình bày phương pháp mà chúng tôi đề nghị để xác định bậc và ước lượng tham số của mô hình ARMA. Trong chương này chúng tôi sẽ xây dựng lại giải thuật tìm kiếm Tabu từ giải thuật tìm kiếm Tabu chuẩn để đưa ra một cơ chế mở rộng không gian tìm kiếm các mô hình ARMA một cách hiệu quả.
- Chương 5 trình bày các kết quả thực nghiệm đạt được từ phương pháp mà chúng tôi đề nghị và đưa ra một số so sánh với các phương pháp trong [5] [28].
- Chương 6 là một số kết luận sau khi thực hiện đề tài.

Chương 2. TỔNG QUAN VỀ PHƯƠNG PHÁP VÀ MÔ HÌNH DỰ BÁO CHUỖI THỜI GIAN

Như đã đề cập sơ qua ở chương 1, việc phân tích chuỗi thời gian nhằm cung cấp những công cụ để lựa chọn mô hình mô tả chuỗi thời gian và có thể sử dụng mô hình cho mục đích dự báo các giá trị tương lai. Tìm mô hình cho chuỗi thời gian là một vấn đề thống kê vì dữ liệu quan sát của chuỗi được sử dụng trong các thủ tục tính toán để ước lượng các hệ số cho mô hình giả thiết. Việc phân loại các phương pháp dự báo tùy thuộc vào mô hình mô hình mà người ta lựa chọn. Đã có rất nhiều chủ đề nghiên cứu về các mô hình dự báo chuỗi thời gian khác nhau [9], trong phần này chúng tôi sẽ điểm sơ qua về các mô hình dự báo chuỗi thời gian thường được biết đến.

Ở giai đoạn đầu của việc nghiên cứu bài toán dự báo chuỗi thời gian, dự báo được thực hiện bằng phương pháp làm trơn và ngoại suy chuỗi dữ liệu thời gian thông qua việc làm khớp toàn cục (*global fit*) trên miền thời gian. Phương pháp này được thay thế bởi sự xuất hiện các mô hình chuỗi thời gian tuyến tính với các đặc điểm nổi trội: rất dễ hiểu để phân tích dữ liệu và rất dễ để thực hiện. Điểm bất lợi của các mô hình này là chúng làm việc không tốt với các chuỗi thời gian được tạo ra bởi các quá trình phi tuyến. Vì lý do đó, các mô hình chuỗi thời gian tuyến tính dần được thay thế trong một chừng mực nhất định bằng các mô hình phi tuyến. Mặc dù áp dụng các mô hình phi tuyến rất thành công với các chuỗi thời gian phức tạp nhưng việc hiểu để giải thích qua các tham số của nó là hết sức khó khăn.

2.1 Các mô hình làm trơn và ngoại suy dữ liệu chuỗi thời gian

2.1.1 Mô hình trung bình di động

Mô hình trung bình di động (*moving average model*) thuộc về lớp các mô hình thường dùng trong dự báo chuỗi thời gian [16]. Giả sử ta cần dự báo chuỗi thời gian được thu thập theo từng tháng trong năm, có thể ta phải dùng đến mô hình sau:

$$f(t) = \frac{1}{12}(y_{t-1} + y_{t-2} + \cdots + y_{t-12}) \tag{2.1}$$

Như vậy, giá trị dự báo 1-bước ứng với mô hình này là:

$$\hat{y}_{T+1} = \frac{1}{12}(y_T + y_{T-1} + \cdots + y_{T-11}) \tag{2.2}$$

Mô hình trung bình di động sẽ hữu dụng nếu ta tin rằng giá trị mong đợi ở tháng kế tiếp của chuỗi thời gian chỉ đơn thuần là giá trị trung bình của 12 tháng trước đó. Điều này có vẻ không thực tế, tuy nhiên, giá trị dự báo tốt có thể đạt được từ việc lấy trung bình đơn giản như vậy. Để hợp lý hơn, ta có thể cho rằng các quan sát gần nhất (với thời điểm dự báo) có vai trò quan trọng hơn là các quan sát trước đó nữa. Trong trường hợp này ta sẽ gán cho các quan sát một hệ số để thể hiện vai trò của nó, quan sát gần nhất sẽ nhận hệ số lớn nhất. Mô hình trung bình di động hoàn thiện theo cách này còn được gọi là *trung bình di động có trọng số theo mũ* (EWMA):

$$\begin{aligned}\hat{y}_{T+1} &= \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \cdots \\ &= \alpha \sum_{\tau=0}^{\infty}(1-\alpha)^{\tau} y_{T-\tau}, 0 < \alpha \leq 1\end{aligned} \tag{2.3}$$

Với $\alpha = 1$, ta bỏ qua bất kỳ quan sát nào xuất hiện trước y_T và giá trị dự báo trở thành:

$$\hat{y}_{T+1} = y_T \tag{2.4}$$

Khi α bé thì mô hình cho thấy các giá trị quan sát càng xa so với thời điểm dự báo càng có vai trò lớn hơn. Chú ý rằng phương trình (2.3) biểu diễn mức trung bình vì

$$\alpha \sum_{\tau=0}^{\infty} (1-\alpha)^{\tau} = \frac{\alpha}{1-(1-\alpha)} = 1$$

Nếu chuỗi thời gian có xu hướng tăng hoặc giảm thì mô hình EWMA sẽ đưa ra giá trị dự báo tương ứng ở mức thấp hơn hoặc cao hơn giá trị tương lai (trường hợp này thực sự có thể xảy ra vì mô hình này lấy trung bình các giá trị trong quá khứ để đưa ra giá trị dự báo, nếu chuỗi thời gian tăng đều đặn thì EWMA sẽ giá trị dự báo bé hơn so với các giá trị của chuỗi gần thời điểm dự báo). Do đó, một kỹ thuật thường thấy trong vấn đề dự báo (không chỉ đối với mô hình EWMA) được áp dụng là loại bỏ các yếu tố xu hướng khỏi dữ liệu chuỗi thời gian trước khi dùng đến mô hình EWMA. Mỗi khi giá trị dự báo của chuỗi đã loại bỏ yếu tố xu hướng được tạo ra thì một số hạng biểu diễn xu hướng sẽ được cộng thêm vào để đạt được giá trị dự báo cuối cùng.

Nếu ta sử dụng mô hình EWMA thực hiện dự báo hơn một bước $\hat{y}_{T+l}$, ta sẽ hiệu chỉnh (2.3) để mở rộng mô hình EWMA như sau:

$$\begin{aligned}\hat{y}_{T+l} = {} & \alpha\hat{y}_{T+l-1} + \alpha(1-\alpha)\hat{y}_{T+l-2} + \cdots + \alpha(1-\alpha)^{l-2}\hat{y}_{T+1} \\ & + \alpha(1-\alpha)^{l-1}y_T + \alpha(1-\alpha)^{l}y_{T-1} + \alpha(1-\alpha)^{l+1}y_{T-2} + \cdots \quad (2.5)\end{aligned}$$

2.1.2 Mô hình làm trơn hàm mũ

Sử dụng mô hình *làm trơn hàm mũ* (*exponential smoothing*) để dự báo có lẽ là phương pháp dự báo được biết đến nhiều nhất [3]. Mô hình san bằng hàm mũ vẫn dựa trên cơ sở của mô hình EWMA, nếu như trong thực tế khi áp dụng EWMA ta chỉ quan tâm đến các quan sát gần với thời điểm dự báo nhất thì mô hình *san bằng hàm mũ đơn giản* (*simple exponential smoothing* - SES) lấy trung bình di động với hệ số giảm dần cho tất cả các quan sát trong quá khứ.

Mô hình san bằng hàm mũ đơn giản được thể hiện bởi phương trình hồi qui sau:

$$\hat{y}_t = \alpha y_t + (1-\alpha)\hat{y}_{t-1} \quad (2.6)$$

Trong đó α ($0 \leq \alpha \leq 1$) là hệ số san bằng, nếu α càng gần 1 thì giá trị hiện tại của y_t càng chiếm phần lớn trong việc sinh ra y_t'. Các giá trị α bé ngụ ý chuỗi được san bằng nhiều hơn, giá trị dự báo mới khá gần với giá trị dự báo cũ và quan sát hiện tại ảnh hưởng rất ít lên giá trị dự báo mới.

Đôi lúc ta muốn san bằng chuỗi thật mạnh nhưng không cho phép các mẫu quá khứ mang trọng số lớn. Trong trường hợp này ta có thể áp dụng *san bằng hàm mũ kép* (DES), tức là ta thực hiện san bằng hàm mũ một lần nữa đối với phương trình (2.6)

$$\hat{\hat{y}}_t = \alpha\hat{y}_t + (1-\alpha)\hat{\hat{y}}_{t-1} \quad (2.7)$$

Theo cách này thì giá trị α lớn có thể được sử dụng.

Ngoài ra còn có *phương pháp làm trơn hàm mũ hai tham số* do Holt đề xuất [7] để dự báo cả giá trị trung bình (như trong mô hình SES) và độ dốc thể hiện xu thế của chuỗi thời gian. Trong mô hình này y_t' được tìm ra từ hai phương trình hồi qui và phụ thuộc vào hệ số san bằng của giá trị trung bình α và hệ số sang bằng của yếu tố xu thế γ, cả hai hệ số này nằm giữa 0 và 1 (α và γ càng nhỏ thì độ mức độ san bằng càng lớn):

$$\hat{y}_t = \alpha y_t + (1-\alpha)(\hat{y}_{t-1} + r_{t-1}) \quad (2.8)$$

$$r_t = \gamma(\hat{y}_t - \hat{y}_{t-1}) + (1-\gamma)r_{t-1} \quad (2.9)$$

Phương trình để dự báo $l - bước$ trong tương lai sẽ là:

$$\hat{y}_{T+l} = \hat{y}_T + lr_T \quad (2.10)$$

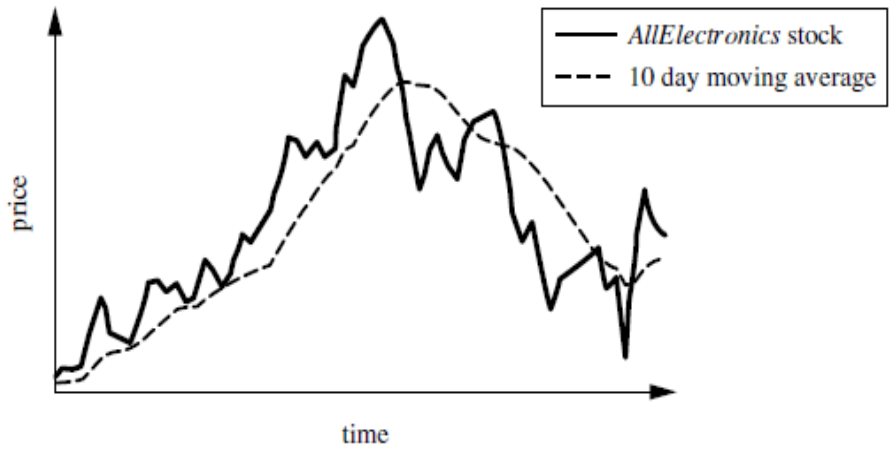
Các phương pháp san bằng có khuynh hướng mò mẫm (còn được gọi là các phương pháp ad-hoc) đặc biệt khi chúng được sử dụng để dự báo. Vấn đề là ta không có cách

xác định giá trị tối ưu nhất cho các hệ số san bằng vì thế việc lựa chọn giá trị thích hợp cho chúng trở nên tùy ý. Nếu mục tiêu chỉ đơn giản là san bằng chuỗi để dễ dàng hơn cho mục đích phân tích thì các phương pháp này không có vấn đề gì vì ta có thể chọn các hệ số san bằng sao cho ta có được mức độ san bằng mong muốn. Tuy nhiên khi sử dụng các mô hình này cho mục đích dự báo thì kết quả dự báo có thể có phần tùy ý.

2.1.3 Dự báo bằng phân tích xu hướng

Như đã đề cập ở các phần trước, hai mục tiêu chính của phân tích chuỗi thời gian là: (1) *xác định mô hình chuỗi thời gian* và (2) *dự báo chuỗi thời gian*. Một trong các phương pháp phân tích được biết đến là phương pháp phân tích xu hướng. Phương pháp này bao gồm 4 thành phần chính đặc thù cho dữ liệu chuỗi thời gian như sau [10]:

- Thành phần xu hướng dài hạn (***T***): thành phần này dùng để chỉ xu hướng tăng hay giảm của đại lượng biểu diễn chuỗi thời gian trong khoảng thời gian dài. Đường cong xu hướng được biểu thị bằng đường nét đứt trong hình 2. Các phương pháp thông thường để xác định đường cong xu hướng là phương pháp *trung bình di động có trọng số* và phương pháp *bình phương cực tiểu*.
- Thành phần chu kỳ (***C***): thành phần này chỉ những dao động dài hạn theo đường cong xu hướng. Những dao động này có thể xuất hiện định kỳ hoặc có thể không. Điều này có nghĩa là các chu kỳ không nhất thiết phải tuân theo chính xác mẫu quan sát tương tự nào đó sau các thời khoảng bằng nhau.
- Thành phần mùa (***S***): thành phần này chỉ sự thay đổi đại lượng biểu diễn chuỗi thời gian y_t theo các mùa trong năm.
- Thành phần bất thường (***I***): thành phần này dùng để chỉ những sự thay đổi bất thường của các giá trị trong chuỗi thời gian. Sự thay đổi này không thể dự đoán bằng các dữ liệu kinh nghiệm trong quá khứ, và về mặt bản chất thành phần này không có tính chu kỳ.



Hình 2.1: Đường cong xu hướng dùng phương pháp trung bình di động

Xác định mô hình chuỗi thời gian được thực hiện bằng cách phân tích chuỗi thời gian thành bốn thành phần như trên. Đại lượng chuỗi thời gian y_t có thể được mô hình để thể hiện mối quan hệ của các thành phần này với nhau bằng cách lấy tích bốn thành phần này.

$$y_t = \mathrm{T.C.S.I} \tag{2.11}$$

2.2 Các mô hình dự báo tuyến tính

Có rất nhiều tài liệu chuyên khảo về các mô hình dự báo tuyến tính, ở đây chúng tôi dựa vào các tài liệu của Weigend và Gershenfeld [1], Box và Jenkin [2] để đưa ra tổng quan về về các mô hình dự báo tuyến tính.

Theo Weigend và Gershenfeld, các mô hình tuyến tính biểu diễn chuỗi thời gian như một tổ hợp tuyến tính của các biến thời gian trễ và có thể có hoặc không có việc kết hợp thêm một đại lượng khác là tổ hợp tuyến tính của các số hạng của quá trình nhiễu trắng. Các đại diện tiêu biểu cho mô hình tuyến tính chẳng hạn như AR, MA và ARMA sẽ lần lượt được trình bày dưới đây.

Mô hình cho một chuỗi thời gian ngẫu nhiên thường được gọi là một *quá trình ngẫu nhiên* (*schochastic process*). Nói cách khác, dữ liệu của bất kỳ chuỗi thời gian nào đều có thể được xem như là được tạo ra nhờ một quá trình ngẫu nhiên. Quá trình ngẫu nhiên có thể được mô tả như một họ các biến ngẫu nhiên được gắn chỉ số thời gian và được ký hiệu bởi $\{X_t,\ t \in T\}$, T là tập các chỉ số thời gian tạo ra quá trình.

2.2.1 Mô hình trung bình di động (MA)

Chuỗi thời gian $\{X_t\}$ được gọi là quá trình trung bình di động bậc q (**MA(q)**) nếu như mỗi quan sát X_t của quá trình **MA(q)** được viết dưới dạng như sau:

$$X_t = \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + \cdots + \theta_q\varepsilon_{t-q} \qquad (2.12)$$

Với $\{\varepsilon_t\}$ là một quá trình *nhiễu trắng* (*white noise*) với trung bình bằng 0, phương sai σ_ε^2 là hằng số và tự hiệp phương sai $\gamma_k = 0$ với $k \neq 0$.

Các quá trình nhiễu trắng thường không thông dụng nhưng một tổ hợp tuyến tính của các số hạng của quá trình nhiễu trắng là một phương pháp tốt để biểu diễn cho các quá trình phi nhiễu trắng.

Phương trình (2.12) cho thấy mô hình MA hoạt động mà không chứa đựng bất kỳ một thông tin phản hồi nào. Có nhiều chuỗi thời gian được làm khớp dựa hoàn toàn trên các thông tin phản hồi, điều này được thực hiện quả mô hình tự hồi qui AR.

2.2.2 Mô hình tự hồi qui (AR)

Đối với mô hình tự hồi qui, chuỗi thời gian $\{X_t\}$ được mô tả bởi phương trình sau:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + \varepsilon_t \qquad (2.13)$$

Phương trình này được gọi là phương trình biểu diễn của mô hình tự hồi qui bậc p (**AR(p)**).

2.2.3 Mô hình ARMA

Nhiều chuỗi thời gian không thể mô hình được như một quá trình trung bình di động hoặc quá trình tự hồi qui thuần túy vì chúng có đặc điểm của cả hai quá trình này. Sử dụng cùng cả hai mô hình AR(p) và MA(q) để mô tả cho các quá trình ngẫu nhiên này tạo ra *mô hình pha trộn tự hồi qui – trung bình di động bậc (p,q)*. Ký hiệu là ARMA*(p,q)* và được biểu diễn như sau:

$$X_t = \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} \qquad (2.14)$$

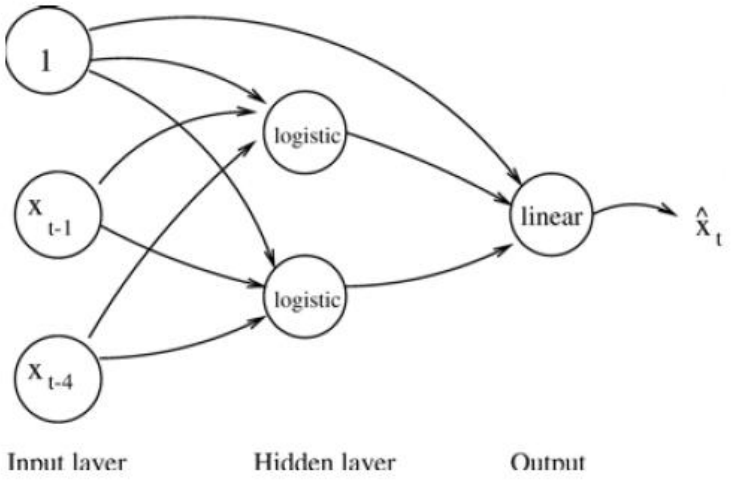
2.3 Các mô hình dự báo phi tuyến

Các mô hình phi tuyến loại bỏ giả thuyết về dữ liệu tuyến tính (hoặc dữ liệu có thể biến đổi bằng một số kỹ thuật để trở nên tuyến tính). Điều này cho phép các mô hình này thích hợp được với rất nhiều loại dữ liệu. Đổi lại, các mô hình phi tuyến cần rất nhiều tham số cũng như không thể giải thích được mô hình dựa trên các tham số đó, chỉ có thể xem mô hình như là một *hợp đen* (*black box*). Với sức mạnh tính toán của máy ngày càng tăng thì số lượng tham số rất nhiều của các mô hình này không còn là vấn đề, ngày nay có rất nhiều mô hình phi tuyến được sử dụng. Ta sẽ khảo sát qua một vài mô hình phi tuyến.

2.3.1 Mạng nơ-ron nhân tạo (ANN)

Mạng nơ-ron nhân tạo (ANN) là một lĩnh vực nghiên cứu rất lớn trong lĩnh vực trí tuệ nhân tạo, ANN được xem như một hệ thống kết nối tập hợp các ngõ vào (inputs) đến tập hợp các ngõ ra (outputs) qua một hay nhiều lớp nơ-ron, các lớp này được gọi là các

lớp ẩn. Việc xác định có bao nhiêu ngõ vào, ngõ ra, số lớp ẩn cũng như là số lượng nơ-ron của mỗi lớp tạo thành kiến trúc của mạng.



Hình 2.2: Kiến trúc của một ANN cho dự báo chuỗi thời gian với 3 ngõ vào, một lớp ẩn hai nơ-ron và một ngõ ra (là giá trị dự báo)

Trong ngữ cảnh chuỗi thời gian, *ngõ ra* là giá trị của chuỗi thời gian được dự báo, *ngõ vào* có thể là có giá trị quan sát trước thời điểm dự báo (xác định bởi độ trễ) của chuỗi thời gian và các biến giải thích khác.

Đối với các ANN một lớp ẩn có H nơ-ron, phương trình tổng quát để tính giá trị dự báo x_t (*ngõ ra*) sử dụng đến các mẫu quan sát quá khứ $x_{t-j_1}, x_{t-j_2}, \ldots, x_{t-j_k}$ làm *ngõ vào* được viết dưới dạng sau:

$$\hat{x}_t = \phi_0 \left(w_{c0} + \sum_{h=1}^{H} w_{h0}\, \phi_h \left(w_{ch} + \sum_{i=1}^{k} w_{ih}\, x_{t-j_i} \right) \right) \qquad (2.15)$$

Trong đó:

- $\{w_{ch}\}_{h=1,2,\ldots,H}$ biểu thị các trọng số cho kết nối giữa hằng số ngõ vào và các nơ-ron lớp ẩn

- w_{c0} là trọng số kết nối trực tiếp giữa ngõ vào hằng số và ngõ ra
- $\{w_{ih}\}$ và $\{w_{h0}\}$ là các trọng số của các kết nối khác giữa các ngõ vào và các nơ-ron lớp ẩn giữa các nơ-ron lớp ẩn với ngõ ra.
- ϕ_0 và ϕ_h là hai hàm kích hoạt lần lượt được sử dụng tại ngõ ra và tại các nơ-ron lớp ẩn.

ANN được áp dụng trong dự báo chuỗi thời gian bởi rất nhiều nhà nghiên cứu [11] [18] [19] [20]. Ở đây ta sẽ điểm qua cách tiếp cận của Wan(1993) [17]. Wan đã hiệu chỉnh thiết kế của mạng nơ-ron chuẩn để mỗi trọng số của mạng và đầu vào là một véctơ thay cho giá trị vô hướng. Véctơ đầu vào mã hóa các giá trị của chuỗi gian. Tích véctơ $w_{ij}^{l}.x_{i}^{l}(k)$ được sử dụng thay cho tích vô hướng (nơ-ron i kết nối với nơ-ron j ở lớp l, k chỉ thời điểm cập nhật trọng số của quá trình lan truyền ngược). Giải thuật lan truyền ngược được tổng quát hóa đối với trường hợp véctơ để huấn luyện mạng này.

Một kiến trúc khác của ANN cho dự báo chuỗi thời gian gọi là *mạng nơ-ron thời gian trễ* [12] [15], trong đó độ trễ thời gian được gắn vào cấu trúc mạng. Phân loại về các kiến trúc mạng nơ-ron cho xử lý chuỗi thời gian có thể xem ở [13]. Các phương pháp này đều gặp phải các vấn đề của một mạng nơ-ron: thời gian huấn luyện lâu, số lượng tham số nhiều. Thực tế, trong trường hợp giải thuật của Wan [17], có 1105 tham số để khớp vào 1000 điểm dữ liệu. Nghĩa là rủi ro về *quá khớp* (*overfitting*) trong quá trình học của mạng là rất lớn.

2.3.2 Các mô hình phi tuyến khác

Mô hình Markov ẩn (HMM) cũng được sử dụng để dự báo dữ liệu chuỗi thời gian [4]. Mô hình Markov ẩn rời rạc không thích hợp để giải quyết các vấn đề liên quan đến dữ liệu liên tục vì vậy một lớp các mô hình HMM được hiệu chỉnh để sử dụng. Thế nhưng mô hình toán học của nó trở nên quá phức tạp để áp dụng thuật toán forward-backward

xác định các tham số. Và do độ phức tạp của giải thuật này là $O(N^2)$ nên rất khó mở rộng cho các tập dữ liệu kích thước lớn.

Cũng có vài phương pháp khác không thông dụng để dự báo phi tuyến. Một trong số đó được gọi *phương pháp Analogues* [8]. Cách tiếp cận này khá đơn giản và chỉ có vài tham số tự do nhưng chỉ áp dụng cho các chu kỳ thời gian ngắn.

2.4 Các công trình liên quan về nhận dạng và ước lượng tham số của mô hình ARMA sử dụng meta-heuristic

Nếu biết trước bậc của mô hình ARMA thì việc ước lượng các tham số để cho mô hình là thích hợp nhất (best fit) với một chuỗi dữ liệu cho trước có thể xem như là đi tìm lời giải cho bài toán tối ưu hóa. Cortez, 2001 đã áp dụng giải thuật di truyền vào việc ước lượng các tham số của mô hình ARMA, trong đó các biến thời gian trễ trong mô hình ARMA được đưa vào dựa trên một chiến lược heuristic là sử dụng các loại *cửa sổ thời gian trượt* STW (*Sliding Time Window*) khác nhau [4]. Có bốn loại STW được đề nghị như sau:

- Loại cửa sổ thời gian trượt đầy đủ với tất cả các biến trễ bắt đầu từ độ trễ 1 cho đến độ trễ tối đa cho trước: STW = <1, 2, …, *m*> (ví dụ *m* có thể được gán bằng 13 là giá trị được xem như là đủ để bao quát các ảnh hưởng như yếu tố mùa vụ (dữ liệu thu thập theo từng tháng) và yếu tố xu hướng).
- Loại cửa sổ thời gian trượt với các biến trễ có hệ số tự tương quan lớn hơn một giá trị ngưỡng nào đó.
- Loại cửa sổ thời gian trượt với bốn biến trễ có hệ số tự tương quan lớn nhất.
- Loại cửa sổ dựa trên phân tích thông tin, chẳng hạn như:
- STW = <1, *K*, *K*+1> nếu chuỗi dữ liệu có yếu tố mùa (chu kỳ *K*) và yếu tố xu thế.
- STW = <1, *K*> nếu chuỗi dữ liệu có yếu tố mùa.

- STW = <1> hoặc STW = <1, 2> nếu chuỗi dữ liệu là xu thế.

Vấn đề xác định bậc của mô hình thường được thực hiện theo phương pháp của Box-Jenkin [2]. Minerva (2001) giới thiệu một phương pháp tính toán sử dụng giải thuật di truyền để lựa chọn một mô hình trong họ các mô hình ARMA. Phương pháp này không chỉ xác định bậc của mô hình ARMA mà còn chỉ ra các biến thời gian trễ có liên quan tham dự vào mô hình [14].

Trong một công trình khác của Cortez (2001), một phương pháp siêu tiến hóa đã được đề xuất trong đó một kiến trúc hai lớp được sử dụng với chức năng của các lớp lần lượt là để tự động hóa quá trình xác định bậc của mô hình ARMA và sau đó ước lượng các tham số của mô hình [5].

Măc dù lớp các mô hình ARMA là những mô hình chỉ phù hợp với các chuỗi dữ liệu thời gian tĩnh (khái niệm chuỗi tĩnh sẽ được trình bày ở chương kế tiếp) nhưng Gnanlet (2009) [23] đã phát triển một meta-heuristics gồm hai giai đoạn dựa trên mô hình ARMA mà không cần giả định về tính tĩnh của chuỗi thời gian. Giai đoạn đầu Gnanlet sử dụng giải thuật mô phỏng luyện kim để xác định bậc tốt nhất của mô hình và giai đoạn hai là sử dụng giải thuật di truyền để xác định các hệ số trong các thành phần AR và MA của mô hình ARMA.

Chương 3. CƠ SỞ LÝ THUYẾT

Trong chương này, chúng tôi sẽ trình bày một số kiến thức cơ bản về chuỗi thời gian liên quan đến mô hình *tự hồi qui kết hợp trung bình di động* – ARIMA (chẳng hạn như khái niệm cơ bản về quá trình ngẫu nhiên và các đặc điểm của chúng, tính tĩnh của một chuỗi thời gian), giải thuật di truyền và các phương pháp xác định bậc và ước lượng các hệ số của mô hình ARMA sử dụng meta-heuristics.

Như đã điểm qua trong 2.4, có nhiều meta-heuristic khác nhau, các meta-heuristic này đều dựa trên một thủ tục chính chạy giải thuật di truyền để ước lượng các hệ số của mô hình ARMA với giả định bậc của mô hình ARMA này đã được xác định trước. Vì lý do đó, trong chương trình này chúng tôi sẽ trình bày cách thức biểu diễn lời giải của giải thuật di truyền cho bài toán ước lượng hệ số của mô hình ARMA. Cuối cùng là chúng tôi trình bày phương pháp meta-heuristic điển hình nhất là phương pháp siêu tiến hóa của Cortez và các cộng sự [5], trong phương pháp này một kiến trúc hai lớp được đề xuất, trong đó lớp thứ nhất (high-level) dùng để giải quyết việc xác định bậc của mô hình và lớp thứ hai (low-level) chính là thủ tục chính chạy giải thuật di truyền để ước lượng các hệ số của mô hình ARMA.

3.1 Các kiến thức cơ bản về chuỗi thời gian

3.1.1 Quá trình ngẫu nhiên

Trong phần này trước hết ta sẽ trình bày vài lý thuyết cơ bản về chuỗi thời gian. Nếu các giá trị trong tương lai của một chuỗi thời gian có thể được dự báo chính xác hoàn toàn từ dữ liệu quá khứ, thì chuỗi thời gian đó được gọi là chuỗi *tất định* (*deterministic*). Tuy nhiên, hầu hết các chuỗi thời gian là *ngẫu nhiên*, nghĩa là trong chuỗi đó giá trị tương lai chỉ được xác định phần nào từ dữ liệu quá khứ. Nếu tìm thấy một mô hình thích hợp để mô tả được hành vi ngẫu nhiên của chuỗi, thì mô hình đó có khả năng là một mô hình dự báo tốt.

Mô hình cho một chuỗi thời gian ngẫu nhiên thường được gọi là một *quá trình ngẫu nhiên* (*schochastic process*). Nói cách khác, dữ liệu của bất kỳ chuỗi thời gian nào đều có thể được xem như là được tạo ra nhờ một quá trình ngẫu nhiên. Quá trình ngẫu nhiên có thể được mô tả như một họ các biến ngẫu nhiên được gắn chỉ số thời gian và được ký hiệu bởi $\{X_t,\ t \in T\}$, T là tập các chỉ số thời gian tạo ra quá trình.

3.1.2 Quá trình ngẫu nhiên tĩnh

Khi phát triển mô hình cho chuỗi thời gian, ta cần biết có *quá trình ngẫu nhiên* cơ bản nào tạo ra chuỗi đó được giả định là *bất biến theo thời gian* hay không. Nếu đặc tính của quá trình ngẫu nhiên thay đổi theo thời gian, tức là quá trình ngẫu nhiên *không tĩnh* (*non-stationary*), thì rất khó biểu diễn chuỗi thời gian trên những thời khoảng quá khứ và tương lai bằng một mô hình đại số đơn giản. Mặt khác, nếu quá trình ngẫu nhiên không đổi theo thời gian, tức là quá trình *tĩnh* (*stationary*), thì ta có thể mô hình quá trình thông qua phương trình với các hệ số cố định và các hệ số này có thể được ước lượng từ dữ liệu quá khứ.

Về mặt toán học một quá trình ngẫu nhiên được gọi là *tĩnh* (ở một số tài liệu khác còn gọi là *tĩnh bậc hai* hay *tĩnh yếu*) nếu như moment bậc một và moment bậc hai của quá trình là hữu hạn và không thay đổi theo thời gian. Moment bậc một là trị trung bình, $E[X_t]$, trong khi moment bậc hai tổng quát là **hiệp phương sai** (**covariance**) giữa X_t và X_{t+k}. Kiểu hiệp phương sai được áp trên cùng một đại lượng X được gọi là **tự hiệp phương sai**. **Phương sai** của quá trình, $Var[X_t]$, là một trường hợp đặc biệt của tự hiệp phương sai khi độ trễ k bằng 0. Do vậy một quá trình có tính chất *tĩnh* nếu như:

1. Trung bình: $E[X_t] = \mu < \infty, \forall t$ (3.1)
2. Phương sai: $Var[X_t] = \sigma^2 < \infty, \forall t$ (3.2)
3. Hàm tự hiệp phương sai chỉ phụ thuộc độ trễ k:
   $Cov[X_t, X_{t+k}] = E[(X_t - \mu)(X_{t+k} - \mu)] = \gamma_k, \forall t$ (3.3)

Tập hợp các hệ số $\{\gamma_k\}, k = 0, 1, 2, ...$ tạo thành hàm tự hiệp phương sai (viết tắt là ACVF) của quá trình. Chú ý rằng $\gamma_0 = \sigma^2$.

ACVF thường được chuẩn hóa để nhận được một tập hợp các ***hệ số tự tương quan*** (**ACF**) $\{\rho_k\}, k = 0, 1, 2, ...$ bởi công thức:

$$\rho_k = \frac{\gamma_k}{\gamma_0} \tag{3.4}$$

Nếu một quá trình ngẫu nhiên là *tĩnh*, phân bố xác suất $p(X_t)$ là như nhau tại mọi thời điểm t và hình dạng của nó (hoặc ít nhất vài đặc điểm của nó) có thể được phỏng đoán bằng cách nhìn vào biểu đồ tần suất của các mẫu quan sát $X_1, X_2, \ldots, X_T$.

Trong thực hành, ước lượng của đại lượng trung bình μ của quá trình có thể đạt được từ *trung bình mẫu* của chuỗi:

$$\bar{X} = \frac{1}{T}\sum_{t=1}^{T} X_t \tag{3.5}$$

Ước lượng cùa phương sai σ^2 có thể tính được từ công thức phương sai mẫu:

$$\hat{\sigma}^2 = \frac{1}{T}\sum_{t=1}^{T}(X_t - \bar{X})^2 \tag{3.6}$$

Và tương tự, ước lượng của hàm tự tương quan cũng được tính từ hàm tự tương quan mẫu:

$$\hat{\rho}_k = \frac{\sum_{t=1}^{T-k}(X_t - \bar{X})(X_{t+k} - \bar{X})}{\sum_{t=1}^{T}(X_t - \bar{X})^2} \tag{3.7}$$

3.1.3 Quá trình không tĩnh thuần nhất

Có lẽ rất ít chuỗi thời gian gặp trong thực tế là chuỗi thời gian tĩnh. Tuy nhiên, thường thì những chuỗi thời gian không tĩnh ta gặp phải thường có tính chất rằng nếu như ta

lấy sai phân một hoặc nhiều lần thì chuỗi kết quả thu được sẽ có tính tĩnh. Chuỗi thời gian không tĩnh thế này được gọi là chuỗi thuần nhất.

Số lần chuỗi gốc phải lấy sai phân được gọi là *bậc thuần nhất*. Do đó, nếu X_t là chuỗi thuần nhất bậc một, thì chuỗi sai phân bậc một sau là chuỗi tĩnh

$$w_t = X_t - X_{t-1} = \Delta X_t$$

Tương tự, chuỗi sai phân bậc hai là:

$$v_t = \Delta X_t - \Delta X_{t-1} = \Delta^2 X_t$$

3.2 Quá trình ARMA

3.2.1 Quá trình trung bình di động

Quá trình ngẫu nhiên $\{X_t\}$ được gọi là quá trình trung bình di động bậc q (**MA(q)**) nếu như nó là một quá trình tĩnh và mỗi quan sát X_t của quá trình **MA(q)** được viết dưới dạng như sau:

$$X_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} \quad (3.8)$$

Với $\{\varepsilon_t\}$ là một quá trình nhiễu trắng với trung bình bằng 0, phương sai σ_ε^2 là hằng số và tự hiệp phương sai $\gamma_k = 0$ với $k \neq 0$.

Mỗi ε_t được giả định sinh ra bởi cùng một quá trình nhiễu trắng vì thế $E[\varepsilon_t] = 0, E[\varepsilon_t^2] = \sigma_\varepsilon^2$ và $E[\varepsilon_t \varepsilon_{t-k}] = 0$ với $k \neq 0$. Từ những kết quả này, ta xác định phương sai của quá trình MA(q) như sau:

$$\begin{aligned} Var[X_t] = \gamma_0 &= E[(X_t)^2] \\ &= E\left(\varepsilon_t^2 + \theta_1^2 \varepsilon_{t-1}^2 + \cdots + \theta_q^2 \varepsilon_{t-q}^2 - 2\theta_1 \varepsilon_t \varepsilon_{t-1} - \cdots\right) \\ &= \sigma_\varepsilon^2 + \theta_1^2 \sigma_\varepsilon^2 + \cdots + \theta_q^2 \sigma_\varepsilon^2 \end{aligned}$$

$$= \sigma_\varepsilon^2\left(1 + \theta_1^2 + \theta_2^2 + \cdots + \theta_q^2\right) \tag{3.9}$$

Từ phương trình (3.9), ta thấy rằng để MA(q) là một quá trình tĩnh thì ta phải có điều kiện sau:

$$\sum_{i=1}^{q} \theta_i^2 < \infty \tag{3.10}$$

Kết quả này là tầm thường vì ta chỉ có hữu hạn các tham số θ_i và dĩ nhiên tổng của chúng là hữu hạn. Tuy nhiên, giả thuyết về số lượng cố định các tham số θ_i có thể được xem xét để có thể xấp xỉ cho mô hình tổng quát hơn. Một mô hình đầy đủ của hầu hết các quá trình ngẫu nhiên yêu cầu vô hạn các số hạng cho tất cả các độ trễ, do đó đối với quá trình MA(∞), ta cũng phải cần có tổng $\sum_{i=1}^{\infty} \theta_i^2$ hội tụ để đảm bảo tính tĩnh cho quá trình MA.

Sự hội tụ sẽ thường xảy ra nếu như các tham số θ_i càng nhỏ khi i càng lớn. Điều này cũng ngụ ý rằng nếu quá trình là tĩnh thì các hệ số tự tương quan ρ_k càng nhỏ khi k càng lớn.

Hàm tự tương quan của một quá trình MA được xác định như sau:

$$\rho_k = \begin{cases} 1 & k = 0 \\ \frac{-\theta_k + \theta_1\theta_{k+1} + \cdots + \theta_{q-k}\theta_q}{1 + \theta_1^2 + \theta_2^2 + \cdots + \theta_q^2} & k = 1, \ldots, q \\ 0 & k > q \end{cases} \tag{3.11}$$

Hàm tự tương quan rất hữu ích để xác định bậc của quá trình MA. Ta có thể xác định bậc q của quá trình MA(q) theo cách sau: ACF sẽ có xu hướng khác không một cách có ý nghĩa thống kê cho đến độ trễ q và sẽ bằng không ngay sau độ trễ q đó.

3.2.2 Quá trình tự hồi qui

Quá trình ngẫu nhiên $\{X_t\}$ được gọi là quá trình tự hồi qui bậc p (**AR(p)**) nếu như nó là một quá trình tĩnh và mỗi quan sát X_t của quá trình **AR(p)** được viết dưới dạng như sau:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + \varepsilon_t \qquad (3.12)$$

Với $\{\varepsilon_t\}$ là một quá trình nhiễu trắng với trung bình bằng 0, phương sai σ_ε^2 là hằng số và tự hiệp phương sai $\gamma_k = 0$ với $k \neq 0$.

Một vấn đề của việc xây dựng mô hình AR là xác định bậc p của quá trình. Mặc dù vài thông tin về bậc của quá trình tự hồi qui có thể đạt được từ hành vi dao động của hàm tự tương quan mẫu, nhưng nhiều thông tin hơn có thể đạt được từ *hàm tự tương quan riêng phần* (PACF).

Để hiểu rõ PACF là gì và cách sử dụng nó như thế nào trước tiên ta hãy xét tự hiệp phương sai và hàm tự tương quan cho một quá trình tự hồi qui bậc p. Tự hiệp phương sai được xác định bởi:

$$\gamma_k = E\left[X_{t-k}\left(\phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + \varepsilon_t\right)\right] \qquad (3.13)$$

Lần lượt thay k = 0, 1, …, p, ta được p+1 phương trình sai phân, giải các phương trình này sẽ xác định được các hệ số γ_0, γ_1, …, γ_k:

$$\begin{aligned}
\gamma_0 &= \phi_1\gamma_1 + \phi_2\gamma_2 + \cdots + \phi_p\gamma_p + \sigma_\varepsilon^2 \\
\gamma_1 &= \phi_1\gamma_0 + \phi_2\gamma_1 + \cdots + \phi_p\gamma_{p-1} \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
\gamma_p &= \phi_1\gamma_{p-1} + \phi_2\gamma_{p-2} + \cdots + \phi_p\gamma_0
\end{aligned} \qquad (3.14)$$

Với $k > p$ thì:

$$\gamma_k = \phi_1\gamma_{k-1} + \phi_2\gamma_{k-2} + \cdots + \phi_p\gamma_{k-p} \tag{3.15}$$

Chia hai vế của hệ (3.14) cho γ_0, ta dẫn ra được p phương trình mà ta có thể xác định p giá trị của ACF:

$$\begin{aligned} \rho_1 &= \phi_1 + \phi_2\rho_1 + \cdots + \phi_p\rho_{p-1} \\ &\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \rho_p &= \phi_1\rho_{p-1} + \phi_2\rho_{p-2} + \cdots + \phi_p \end{aligned} \tag{3.16}$$

Với $k > p$ ta suy ra ρ_k từ phương trình (3.11):

$$\rho_k = \phi_1\rho_{k-1} + \phi_2\rho_{k-2} + \cdots + \phi_p\rho_{k-p} \tag{3.17}$$

Hệ phương trình (3.17) được gọi là hệ phương trình Yule-Walker. Ta trình bày hệ phương trình này dưới dạng sau:

$$\begin{bmatrix} 1 & \rho_1 & \cdots & \rho_{p-1} \\ \rho_1 & 1 & & \rho_{p-2} \\ \vdots & & \ddots & \vdots \\ \rho_{p-1} & \rho_{p-2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_p \end{bmatrix} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_p \end{bmatrix} \tag{3.18}$$

Nếu biết ρ_1, ρ_2, …, ρ_p thì ta sẽ giải hệ để xác định ϕ_1, ϕ_2, …, ϕ_p. Không may là lời giải của hệ phương trình Yule-Walker lại đòi hỏi cần phải biết trước bậc p của quá trình tự hồi qui. Vì thế, ta giải hệ Yule-Walker ứng với các giá trị p liên tiếp nhau. Nói cách khác, ta bắt đầu với giả thiết rằng p = 1, hệ (3.16) sẽ trở thành $\rho_1 = \phi_1$ hoặc sử dụng tự tương quan mẫu $\hat{\rho}_1 = \hat{\phi}_1$. Do vậy, nếu giá trị tính toán được $\hat{\phi}_1$ khác 0 đáng kể thì ta biết rằng quá trình AR có bậc ít nhất bằng 1. Ta ký hiệu giá trị $\hat{\phi}_1$ là a_1.

Tiếp tục xét giả thiết p = 2 và giải hệ (3.16) với trường hợp p = 2 sẽ cho ta tập các giá trị mới được ước lượng $\hat{\phi}_1$ và $\hat{\phi}_2$. Nếu $\hat{\phi}_2$ khác 0 đáng kể thì ta biết rằng quá trình AR

có bậc ít nhất bằng 2, trong khi nếu như $\hat{\phi}_2$ xấp xỉ 0 thì ta kết luận răng p = 1. Ta ký hiệu giá trị $\hat{\phi}_2$ là a_2.

Ta lặp lại quá trình này cho các giá trị p kế tiếp. Với p = 3, ta đạt được ước lượng $\hat{\phi}_3$, và ký hiệu là a_3, … Ta gọi chuỗi a_1, a_2, a_3, ... là PACF và từ chuỗi này ta có thể phỏng đoán bậc của quá trình tự hồi qui. Đặc biệt nếu bậc của quá trình thực sự là *p*, ta quan sát thấy rằng $a_j \approx 0,\ j > p$.

3.2.3 Quá trình ARMA

Nhiều quá trình ngẫu nhiên tĩnh không thể mô hình được như một quá trình trung bình di động hoặc quá trình tự hồi qui thuần túy vì chúng có đặc điểm của cả hai quá trình này. Sử dụng cùng cả hai mô hình AR(p) và MA(q) để mô tả cho các quá trình ngẫu nhiên này tạo ra *mô hình pha trộn tự hồi qui – trung bình di động bậc (p,q)*. Ký hiệu là ARMA*(p,q)* và được biểu như sau:

$$X_t = \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} \qquad (3.19)$$

Điều kiện cần cho tính tĩnh của quá trình là:

$$\phi_1 + \phi_2 + \cdots + \phi_p < 1$$

Mô hình ARIMA cho quá trình không tĩnh thuần nhất:

Trong thực tế, ta phải làm việc với rất nhiều chuỗi không tĩnh, vì thế các tính chất của quá trình ngẫu nhiên thay đổi theo thời gian. Trong phần này ta sẽ trình bày cách thức áp dụng mô hình cho các chuỗi không tĩnh, các chuỗi này có thể biến đổi về chuỗi tĩnh bằng cách lấy sai phân một hoặc nhiều lần. Ta nói rằng X_t là *một quá trình phi tĩnh thuần nhất bậc d* nếu

$$w_t = \Delta^d X_t \qquad (3.20)$$

là một chuỗi tĩnh. Ở đây Δ là ký hiệu phép toán lấy sai phân, chẳng hạn như:

$$\Delta X_t = X_t - X_{t-1} \text{ hay } \Delta^d X_t = \Delta X_t - \Delta X_{t-1}$$

Sau khi lấy sai phân chuỗi X_t để tạo ra chuỗi tĩnh w_t, ta có thể mô hình w_t như một quá trình ARMA.

Nếu $w_t = \Delta^d X_t$ và w_t là một quá trình ARMA(p,q) thì ta nói rằng X_t là một quá trình *tự hồi qui kết hợp trung bình trượt có bậc (p,d,q)* hay đơn giản là quá trình ARIMA(p,d,q).

Vấn đề lựa chọn mô hình ARIMA

Ta đã thấy rằng bất kỳ chuỗi thời gian không tĩnh thuần nhất nào cũng có thể được mô hình thành quá trình ARIMA(p,d,q). Vấn đề thực tế là lựa chọn các giá trị p, d, và q phù hợp nhất để chỉ ra mô hình ARIMA. Vấn đề này phần nào được giải quyết bằng cách khảo sát đồng thời hàm tự tương quan và hàm tự tương quan riêng phần đối với chuỗi thời gian cần xem xét.

Qui trình để lựa chọn mô hình ARIMA thường diễn ra như sau:

- Kiểm tra xem chuỗi dữ liệu có tính tĩnh hay không? Nếu không tĩnh thì xác định số lần lấy sai phân d
- Khi đã chuyển sang chuỗi tĩnh, bước quan trọng kế tiếp là xác định p và q
- Đối với mô hình MA(q) thuần túy, ACF sẽ có xu hướng khác không đáng kể cho đến độ trễ q và sẽ bằng 0 ngay sau độ trễ q đó. Trong khi đó, PACF sẽ có xu hướng bằng 0 ngay lập tức.
- Đối với mô hình AR(p) thuần túy, ACF sẽ có xu hướng bằng 0 ngay lập tức, trong khi đó PACF sẽ có xu hướng khác không đáng kể cho đến độ trễ p và sẽ bằng không ngay sau độ trễ p đó.

Nếu cả *p* và *q* đều khác không, ta sẽ áp dụng mô hình pha trộn ARMA cho chuỗi dữ liệu đã qua biến đổi sang chuỗi tĩnh. Trong trường hợp này khó xác định chính xác số bậc của AR và MA, nên ta phải sử dụng nhiều mô hình khác nhau để tiến hành so sánh và lựa chọn.

3.3 Giải thuật di truyền

Giải thuật di truyền (*genetic algorithm* - ***GA***) là một kỹ thuật của khoa học máy nhằm tìm kiếm giải pháp thích hợp cho các bài toán tối ưu tổ hợp (*combinatorial optimization*). Giải thuật này lần đầu tiên được đề xuất bởi Holland, 1975 dựa trên ý tưởng áp dụng các nguyên lý tiến hóa trong tự nhiên như di truyền, đột biến, chọn lọc tự nhiên và trao đổi chéo [21]. Cụ thể, ý tưởng của giải thuật di truyền được tóm lược như sau: *Trong một quần thể sẽ tồn tại nhiều cá thể, trong đó có những cá thể khỏe mạnh và những cá thể yếu kém. Trong quá trình sinh sống, dưới tác động của điều kiện ngoại cảnh, các cá thể yếu kém sẽ dần mất đi, các cá thể khỏe manh thì được giữ lại. Đây chính là ý niệm cơ bản của quá trình chọn lọc tự nhiên. Các cá thể được giữ lại sẽ thực hiện quá trình sinh sản tạo ra các cá thể con có thể tốt hơn nếu như mang được các đặc tính tốt của cả bố và mẹ. Đôi khi một vài cá thể lại bị đột biến gen sẽ giúp cho quần thể có thêm những đặc tính mới. Dĩ nhiên trong quá trình sinh sản cũng như đột biến, có thể các cá thể yếu kém sẽ được tao ra, nhưng những cá thể như vậy sẽ dần bị loại bỏ qua quá trình chọn lọc tự nhiên.* Giải thuật di truyền mô phỏng quá trình này sẽ xem xét lời giải của bài toán cần giải quyết như một cá thể, sau đó thực hiện chọn lọc, lai, đột biến quần thể lời giải, sau một số thế hệ nhất định sẽ cho lời giải chấp nhận được.

Giải thuật di truyền cho rằng quá trình tiến hóa tự nhiên là quá trình hoàn hảo nhất, hợp lý nhất, và tự nó đã mang tính tối ưu. Quan điểm này được coi như là một tiên đề đúng, không chứng minh được, nhưng phù hợp với thực tế khách quan. Quá trình tiến hóa thể hiện tính tối ưu ở chỗ, thế hệ sau được tạo ra tốt hơn thế hệ trước, và nếu thực hiện tiến

hóa đến một số lượng thế hệ nhất định nào đó ta sẽ được những cá thể đạt độ tốt như ta mong muốn.

```
begin
   KHỞI TẠO ngẫu nhiên quần thể các cá thể;
   ƯỚC LƯỢNG hàm thích nghi cho từng cá thể;
   repeat
      CHỌN LỌC các cá thể cha mẹ;
      LAI GHÉP các cặp cá thể cha mẹ dựa vào xác suất lai ghép;
      ĐỘT BIẾN cho các các thể con dựa vào xác suất tao đột biến;
      ƯỚC LƯỢNG các cá thể con;
      CHỌN LỌC các cá thể cho thế hệ kế tiếp
   until ĐIỀU KIỆN DỪNG được thỏa mãn
end
```

Hình 3.1: Chi tiết hoạt động của một giải thuật di truyền chuẩn

Một giải thuật di truyền có các thành phần cơ bản sau:

- Mỗi một lời giải của bài toán sẽ được thể hiện bởi một nhiễm sắc thể (NST)
- Cách khởi tạo quần thể ban đầu (tập các NST)
- Định nghĩa hàm thích nghi để đánh giá độ tốt xấu của NST
- Các phép toán di truyền: chọc lọc, lai, đột biến
- Các tham số của giải thuật: số dân (hay kích thước của quần thể), xác suất lai ghép, đột biến.
- Điều kiện dừng của giải thuật

3.3.1 Cách biểu diễn di truyền cho lời giải của bài toán

Khi áp dụng giải thuật di truyền để giải một bài toán, mỗi lời giải của bài toán được coi như là một cá thể trong quần thể (tập hợp các lời giải). Chúng ta phải thực hiện quá trình tiến hóa để tìm ra lời giải tối ưu. Mỗi lời giải hay mỗi cá thể được biểu diễn dưới dạng một nhiễm sắc thể (NST). Mỗi NST là một chuỗi các gen. Có nhiều cách để biểu diễn NST và gen tùy thuộc vào từng bài toán cụ thể. Sau đây là một số phương pháp biểu diễn:

Biểu diễn nhị phân: Mỗi gen được biểu diễn bởi một bit nhị phân (nhận giá trị 0 hoặc 1) và một NST là một chuỗi các gen có độ dài là n.

Biểu diễn sử dụng hoán vị: Mỗi NST tương ứng với một hoán vị của một tập n ký hiệu.

Biểu diễn bằng giá trị: Biểu diễn NST trực tiếp bằng các gen có giá trị số thực.

3.3.2 Cách khởi tạo quần thể ban đầu

Việc khởi tạo quần thể thường được thực hiện khá đơn giản bằng cách khởi tạo ngẫu nhiên. Tuy nhiên với một số bài toán, ta có thể sử dụng các kỹ thuật để khởi tạo quần thể ban đầu tốt hơn, giúp giải thuật tăng tốc quá trình tiến hóa và cho kết quả tốt hơn. Chẳng hạn như theo phương pháp Decimation, quần thể ban đầu được tạo ra bằng cách chọn ra $\boldsymbol{n}$ cá thể tốt nhất từ tập $\boldsymbol{n.d}$ các cá thể được tạo ra ngẫu nhiên. Hệ số $\boldsymbol{d}$ được gọi là hệ số *decimatino* [30].

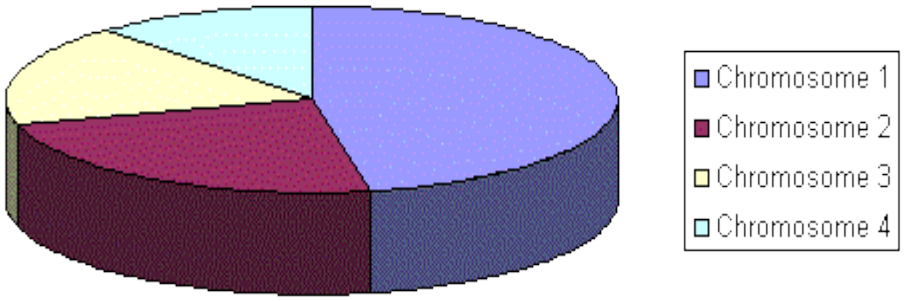
3.3.3 Phép toán chọn lọc

Là một quá trình mà trong đó các NST được lựa chọn tùy thuộc vào giá trị độ thích nghi. Các NST được chọn lọc từ quần thể để làm cha mẹ trong quá trình lai tạo hoặc NST được chọn để giữ lại trong thế hệ tiếp theo. Vấn đề là chọn các NST như thế nào?

Theo lý thuyết tiến hóa của Darwin thì những cá thể tốt hơn sẽ có cơ hội sống và lai tạo nhiều hơn. Có nhiều phương pháp để lựa chọn NST tuân theo qui luật này và một số phương pháp phổ biến sẽ được trình bày dưới đây:

Chọn lọc dùng bánh xe Roulette (Roulette Wheel Selection)

Trong phương pháp này, các cá thể cha mẹ được lựa chọn theo độ thích nghi của chúng. NST có độ thích nghi càng lớn thì cơ hội lựa chọn càng cao. Ta hãy tưởng tượng một **bánh xe Roulette** đặt tất cả các NST của quần thể vào trong đó, mỗi NST sẽ có vị trí của nó trên bánh xe Roulette tùy vào giá trị của hàm thích nghi của NST đó, kết quả của mỗi lần quay bánh xe sẽ tương ứng với một NST được chọn.



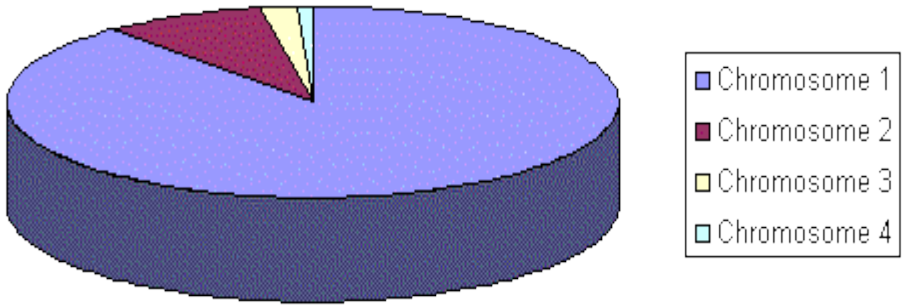
Hình 3.2: Minh họa bánh xe Roulette

Thủ tục thực hiện bánh xe Roulette qua các bước sau đây:

1. Lấy tổng - Tính tổng **S** của tất cả các độ thích nghi của các cá thể trong quần thể
2. Lựa chọn - Tạo một số ngẫu nhiên ***r*** trong khoảng (0,S)
3. Bước lặp - Duyệt qua từng cá thể trong quần thể theo thứ tự từ cá thể đầu tiên, mỗi lần duyệt đến cá thể nào thì cộng dồn độ thích nghi của cá thể đó vào lại với nhau, đặt là giá trị **s**. Khi gặp cá thể nào làm cho **s** > ***r*** thì dừng quá trình duyệt tuần tự và trả về cá thể đó.

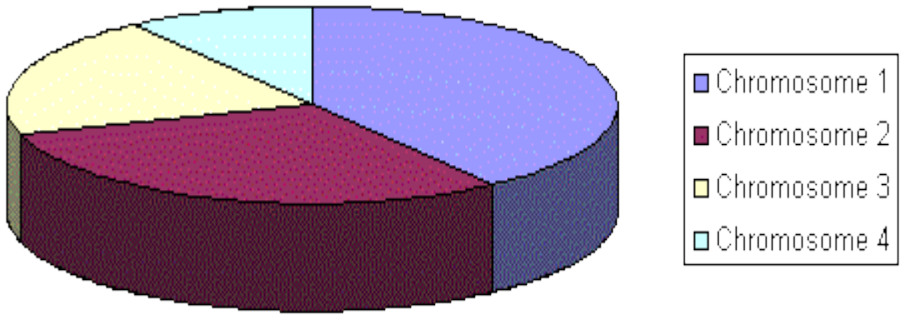
Chọn lọc xếp hạng (Rank Selection)

Phương pháp chọn lọc dùng bánh xe Roullete vấp phải vấn đề khi các giá trị độ thích nghi chênh lệnh nhau quá lớn, chẳng hạn như nếu độ thích nghi cao nhất chiếm 90% tổng giá trị độ thích nghi thì hầu như chỉ có NST có độ thích nghi cao nhất này được chọn để lai tạo và giữ lại ở thế hệ kế tiếp, các NST khác sẽ có rất ít cơ hội được lựa chọn.



Hình 3.3: Tình huống trước khi xếp hạng

Phương pháp chọn lọc xếp hạng sẽ khắc phục vấn đề này bằng cách chọn các NST dựa vào thứ tự xếp hạng của nó, chứ không dựa vào giá trị độ thích nghi. Đầu tiên mỗi NST sẽ được gán một vị trí trong bảng xếp hạng, sau đó có thể áp dụng bánh xe Roullete để chọn lọc dựa trên bảng xếp hạng.



Hình 3.4: Bánh xe Roullete của quần thể sau khi đã xếp hạng

Elitism

Khi tạo ra thế hệ mới các NST bằng cách phép toán lai và đột biến (sẽ trình bày ở những phần kế tiếp), chúng ta sẽ đứng trước một sự thay đổi lớn đó là ta sẽ mất đi NST tốt nhất. **Elitism** là phương pháp lưu giữ bản sao của NST tốt nhất qua từng thế hệ giúp giải thuật di truyền ngăn chặn việc làm mất đi lời giải tốt nhất của bài toán.

3.3.4 Phép toán lai

Phép toán lai là quá trình hình thành NST mới từ các NST cha mẹ bằng cách ghép một hoặc nhiều đoạn gen của cả hai NST cha mẹ với nhau.

Lai ghép một điểm (one-point crossover)

Đây là cách lai ghép đơn giản nhất. Đầu tiên, một vị trí được chọn ngẫu nhiên trên hai chuỗi gen được gọi là điểm lai, sau đó các chuỗi này được tiến hành ghép chéo nhau tại vị trí này. Quá trình này sẽ tạo ra hai chuỗi mới, mỗi chuỗi mới sẽ được lấy từ phần bên phải của chuỗi cha ghép với phần bên trái cuỗi chuỗi mẹ tính từ vị trí ghép chéo và thực hiện ngược lại một cách tương tự cho chuỗi còn lại. Ví dụ chúng ta lai ghép cặp cha mẹ như sau:

Parent1: 7 3 | 7 6 1 3
Parent2: 1 7 **|** 4 5 2 2

Kết quả lai ghép sẽ là:

Child 1: 7 3 | 4 5 2 2
Child 2: 1 7 | 7 6 1 3

Lai ghép hai điểm (two-point crossover)

Tương tự như phép lai ghép một điểm, thay vì chọn một điểm, chúng ta chọn hai điểm cho vị trí ghép và hoán vị phần giữa hai điểm lai của hai chuỗi gen cha mẹ với nhau. Ví dụ với hai NST 737613 và 174522, chúng ta chọn ngẫu nhiên 2 điểm lai, giả sử là 2 và 4:

Parent1: 7 3 | **7 6** | 1 3
Parent2: 1 7 **| 4 5** | 2 2

Kết quả lai ghép sẽ là:

Child 1: 7 3 | **4 5** | 1 3
Child 2: 1 7 | **7 6** | 2 2

Lai ghép đều (uniform crossover)

Mỗi gen của NST cha có xác suất hoán đổi với gen ở vị trí tương ứng của NST mẹ là 0.5. Lai ghép đều được thực hiện như sau: với mỗi gen ta sinh ra một số ngẫu nhiên ***r*** trong khoảng (0,1). Nếu ***r*** > 0.5 thì hoán đổi hai gen, ngược lại thì giữ nguyên. Ví dụ với 2 NST 737613 và 174522, các xác suất tương ứng ở các vị trí gen là 0.2, **0.7**, 0.3, 0.4, **0.8** và 0.1 thì kết quả của phép toán lai ghép đều như sau:

Parent1: 7 **3** 7 6 **1** 3
Parent2: 1 **7** 4 5 **2** 2

Kết quả lai ghép sẽ là:

Child 1: 7 **7** 4 5 **2** 3
Child 2: 1 **3** 7 6 **1** 2

3.3.5 Phép toán đột biến

Mặc dù phép lai ghép sản sinh ra nhiều chuỗi mới nhưng nó không giới thiệu các thông tin mới nào trong quần thể ở cấp độ bit. Phép toán đột biến sẽ đưa ra những thông tin mới trong quần thể. Với một chuỗi các bit mới, ta áp dụng đột biến với một xác suất thấp P_m, nó có tác dụng thực hiện phép đảo bit đối với một gen được lựa chọn ngẫu nhiên.

- Biểu diễn nhị phân: Chọn một số bit rồi đảo giá trị các bit đó, hoặc chọn hai bit khác nhau bất kỳ sau đó hoán đổi chúng với nhau: (0 1 **1** 0 1) => (0 1 **0** 0 1)
- Biểu diễn bằng hoán vị: Chọn hai vị trí bất kỳ rồi hoán đổi giá trị của chúng với nhau: (1 2 **3** 4 5 6 7 **8** 9) => (1 2 **8** 4 5 6 7 **3** 9)

3.3.6 Các tham số của giải thuật

- Kích thước (hay dân số) của quần thể (POP-SIZE): giá trị thể hiện có bao nhiêu NST trong quần thể. Việc lựa chọn kích thước quần thể là quan trọng, phải có tính cân nhắc và thường được căn cứ vào quá trình làm thực nghiệm của mỗi bài toán. Nếu chọn kích thước nhỏ thì giải thuật chỉ tìm kiếm lời giải trong một khoảng nhỏ của không gian tìm kiếm, như vậy dẫn đến trường hợp kết quả đạt được không tối ưu, nhưng nếu chọn kích thước lớn thì chương trình sẽ chạy chậm.
- Xác suất lai ghép $\mathbf{P_c}$ cho biết việc lai ghép có được thực hiện thường xuyên hay không. Không phải lúc nào việc lai ghép giữa hai cá thể cha mẹ cũng cho cá thể con tốt hơn. Do đó việc chọn xác suất lai ghép cao không phải lúc nào cũng tốt.
- Xác suất tạo đột biến $\mathbf{P_c}$ thường chọn giá trị nhỏ, nếu quá lớn thì giải thuật di truyền sẽ hoạt động không khác gì một giải thuật tìm kiếm ngẫu nhiên.

3.3.7 Điều kiện dừng của giải thuật

Một số tiểu chuẩn thường dùng để dừng giải thuật di truyền:

- Vượt quá giới hạn thời gian tính toán cho phép
- Tổng độ thích nghi của quần thể đạt tới giới hạn cho phép
- Giá trị NST tốt nhất có độ thích nghi đạt tới ngưỡng đặt ra
- Thế hệ tiến hóa vượt quá số thế hệ tối đa của quá trình tiến hóa

3.4 Mô hình ARMA sử dụng giải thuật di truyền

Trong phần này, chúng tôi trình bày mô hình ARMA có thể được biểu diễn bằng một nhiễm sắc thể (thành phần biểu diễn nghiệm) trong giải thuật di truyền như thế nào. Mỗi nhiễm sắc thể mã hóa lời giải thành một chuỗi các giá trị nhị phân, ứng mỗi mỗi nhiễm sắc thể sẽ được gán một giá trị đánh giá chất lượng của lời giải thông qua *hàm thích nghi*. Các lời giải mới sẽ được tạo ra thông qua việc áp dụng các toán tử *lai ghép* và *đột biến*. Toàn bộ quá trình là một quá trình chọn lọc tự nhiên cho đến khi tìm ra được mô hình ARMA phù hợp nhất với với chuỗi dữ liệu (tức là nhiễm sắc thể biểu diễn cho mô hình ARMA được đánh giá tốt nhất bởi hàm thích nghi).

Sau đây là cách thức ánh xạ mô hình ARMA thành một nhiễm sắc thể như thế nào để có thể sử dụng giải thuật di truyền nhằm mục đích ước lượng tham số của mô hình.

3.4.1 Ánh xạ mô hình ARMA thành nhiễm sắc thể

Theo công trình của Minerva (2001) và các cộng sự [14], một mô hình ARMA là một nhiễm sắc thể trong quần thể các lời giải của giải thuật di truyền có thể được mã hóa dưới dạng một chuỗi các giá trị nhị phân 0 và 1 để biểu diễn các bậc khác nhau và các tham số khác nhau của mô hình. Mỗi nhiễm sắc thể là sự kết hợp của nhiều phần khác nhau để nhận diện được:

- *P* số hạng của quá trình tự hồi qui
- *Q* số hạng của quá trình trung bình di động (lưu ý rằng *P*, *Q* ở đây không thể hiện bậc của mô hình)

- các số hạng X_i của thành phần AR
- các số hạng ε_j của thành phần MA

Cụ thể trong cách biểu diễn nhiễm sắc thể này, ta giả sử rằng hai phần đầu tiên của nhiễm sắc thể mỗi phần bao gồm 4 chữ số nhị phân mã hóa số nguyên chạy từ 1 đến 15 nhằm lần lượt lựa chọn *P* số hạng của quá trình tự hồi qui và *Q* số hạng của quá trình trung bình di động. *P+Q* phần kế tiếp của nhiễm sắc thể mỗi phần bao gồm 5 chữ số nhị phân . Mỗi phần này mã hóa số nguyên nhị phân chạy từ 1 đến 31 biểu diễn số hạng có độ trễ tương ứng với số nguyên này của các thành phần AR và MA. Ví dụ giả sử ta có chuỗi bit biểu diễn nhiễm sắc thể như sau:

$$\underbrace{0010}_{P}\ \underbrace{0001}_{Q}\ \underbrace{00001\ 10110}_{P \text{ số hạng trễ của AR}}\ \underbrace{01000}_{Q\ số\ hạng\ trễ\ của\ MA}$$

Tương ứng với giá trị thực của:

$$\underbrace{2}_{P}\ \underbrace{1}_{Q}\ \underbrace{1}_{X_{t-1}}\ \underbrace{22}_{X_{t-22}}\ \underbrace{8}_{\varepsilon_{t-8}}$$

Tức là nhiễm sắc thể này mô hình cho chuỗi thời gian X_t có hai số hạng tự hồi qui, 1 số hạng trung bình di động và phương trình cho mô hình này là:

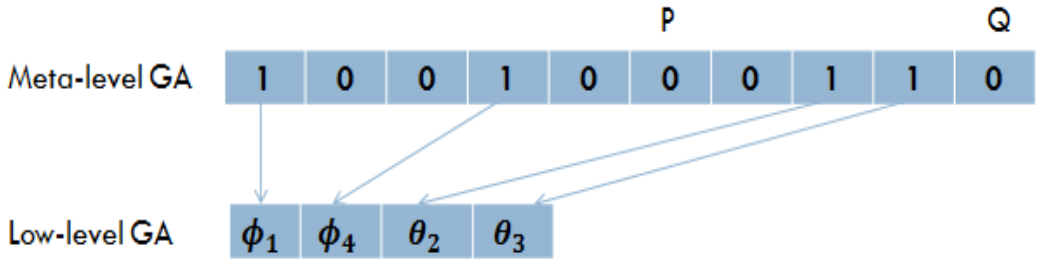
$$X_t - \phi_1 X_{t-1} - \phi_{22} X_{t-22} = \varepsilon_t - \theta_{t-8}$$

Để xây dựng giải thuật di truyền, ta sinh ra ngẫu nhiên một quần thể các nhiễm sắc thể biểu diễn các mô hình ARMA ứng viên của chuỗi thời gian cụ thể.

Với mỗi mô hình trong quần thể, ta thực hiện ước lượng thống kê cho các tham số của mô hình sử dụng các mẫu quan sát của chuỗi thời gian.

3.4.2 Phương pháp siêu tiến hóa cho mô hình ARMA

Để thay thế cho các phương pháp thử sai truyền thống trong việc lựa chọn mô hình ARMA tốt nhất vì những hạn chế của phương pháp này khi không gian tìm kiếm của bài toán là rất lớn, Cortez và các cộng sự (2001) đã đề xuất một *phương pháp siêu tiến hóa* (*meta-evolutionary approach*) cho mô hình ARMA [4]. Thông thường, các meta-GA được sử dụng để tối ưu các thông số của GA (chẳng hạn như kích thước quần thể, hệ số tạo đột biến tức xác suất để thực hiện tác vụ đột biến của GA), trong phương pháp của Cortez, một kiến trúc hai lớp được đề nghị, lớp bên trên có nhiệm vụ chọn lựa mô hình được gọi là *meta-level GA*, lớp bên dưới có nhiệm vụ ước lượng tham số cho mô hình được gọi là *low-level GA*.



Hình 3.5: Minh họa cho việc giải mã của một nhiễm sắc thể trong meta-level

Các nhiễm sắc thể dùng trong meta-level GA là một chuỗi bit nhị phân, mỗi nhiễm sắc thể mã hóa cho duy nhất một mô hình ARMA, mỗi một *gene* (trường hợp cụ thể này là một bit) trong nhiễm sắc thể cho biết rằng có xuất hiện hay không hệ số tương ứng của mô hình.

Theo Cortez và các cộng sự (2001) [5], để biểu diễn các nhiễm sắc thể của low-level GA, mô hình *ARMA(P,Q)* có thể viết dưới dạng sau:

$$\hat{X}_t = g_0 + \sum_{i\in\{1,\dots,P\}} g_i X_{t-k_i} + \sum_{j\in\{1,\dots,Q\}} g_{j+P}\varepsilon_{t-k_j} \quad (3.21)$$

Với g_i thay thế cho *gene thứ i* của nhiễm sắc thể biểu diễn các hệ số của mô hình ARMA, dãy $<k_1,k_2,\ldots,k_n>$ là *cửa sổ thời gian trượt* (*sliding time window*).

Ở low-level này, giải thuật di truyền sử dụng các *nhiễm sắc thể biểu diễn giá trị trị thực* (*Real-Valued Representation* hay *floating point chromosome representation* ở một số tài liệu khác), mỗi *gen* của nhiễm sắc thể mã hóa một hệ số của mô hình ARMA như trong phương trình (3.21). Mỗi cá thể (tức mỗi nhiễm sắc thể bên trong quần thể) được đánh giá bởi *RMSE* (*Root Mean Square*) trên toàn bộ tập huấn luyện.

$$RMSE = \sqrt{\frac{SSE}{l}} \tag{3.22}$$

$$SSE = \sum_{i=1}^{l} e_i^2 \tag{3.22}$$

$$e_t = X_t - \hat{X}_t \tag{3.23}$$

Với:

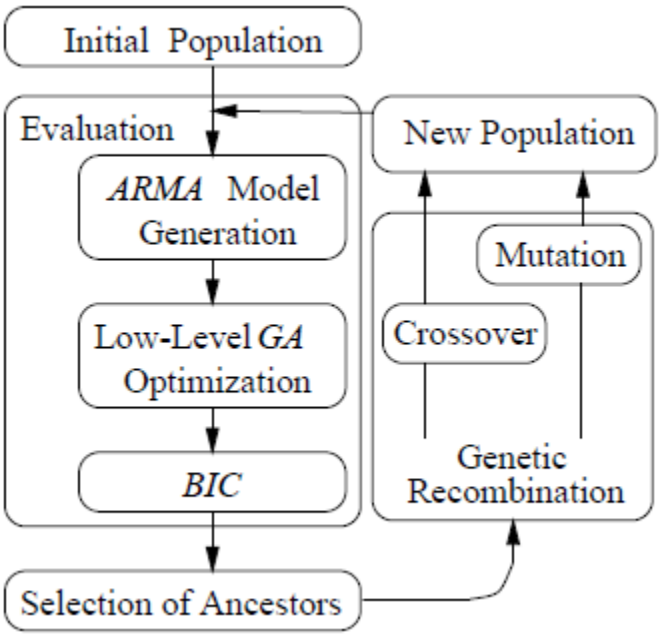
- e_t là sai số dự báo
- *SSE* (*Sum Square Error*) là tổng sai số bình phương
- *l* là số lượng dự báo

Nói cách khác, độ thích nghi của mỗi cá thể trong meta-level GA được đo bằng cách giải mã nhiễm sắc thể từ meta-level sang low-level, thực hiện GA trên đó và cuối cùng tính giá trị *BIC* (*Bayesian Information Criterion*) [22] trên mô hình được tối ưu đó.

$$BIC = N.\ln\left(\frac{\text{SSE}}{N}\right) + p.\ln(N) \tag{3.24}$$

Với *N* là số lượng mẫu huấn luyện và p là số lượng tham số của mô hình.

Toàn bộ hệ thống của phương pháp siêu tiến hóa được mô tả trong hình 3.6.



Hình 3.6: Phương pháp siêu tiến hóa

Dựa trên những kiến thức vừa được trình bày, trong chương kế tiếp, chúng tôi sẽ đề xuất một phương pháp mới để mở rộng không gian tìm kiếm lời giải cho mô hình ARMA mà không dựa vào phép toán lai ghép ngẫu nhiên như trong mức meta-level của phương pháp siêu tiến hóa của Cortez. Hơn nữa, thủ tục chính sử dụng giải thuật di truyền để xác định hệ số của mô hình ARMA (chính là mức low-level của phương pháp siêu tiến hóa) sẽ được chúng tôi đề xuất bằng cách kết hợp các biến thể khác nhau của các phép toán lai ghép và phép toán đột biến dựa trên những tổng kết gần đây về các heuristic được áp dụng trong các bài toán tối ưu số (*numerical optimization*) [25].

Chương 4. PHƯƠNG PHÁP GIẢI QUYẾT VẤN ĐỀ

Như đã trình bày trong chương 3, Cortez đã sử dụng các giải thuật di truyền ở hai mức khác nhau để xác định bậc và ước lượng tham số của mô hình ARMA [4][5]. Tuy nhiên việc sử dụng một kiến trúc hai mức như vậy trong đó mức meta-level đảm nhận việc tìm kiếm các mô hình ARMA tốt hơn sử dụng giải thuật di truyền sử dụng phép toán lai hai điểm không đem lại nhiều ý nghĩa trong việc lựa chọn mô hình dù rằng hàm thích nghi được sử dụng để đánh giá là đại lượng BIC, tuy nhiên bản chất vẫn là sự hoán đổi các bậc của hai NST cha mẹ một cách ngẫu nhiên. Ngoài ra giải thuật siêu tiến hóa do Cortez đề xuất chạy quá chậm.

// HIGH-LEVEL	// LOW-LEVEL

Select an initial solotion s_0
Select an initial temperature $t = t_0 > 0$
Select maximum number of phase *maxphase*
Select a temperature reduction coefficient α
while *phase* < *maxphase*
  while *iteration_count* < *nrep*
  /* *s* is a neighbor solution of s_0 */
  Randomly select $s \in N(s_0)$
  /* compute the change in cost function */
  $\delta = f(s) - f(s_0)$
  if $\delta < 0$ **then** $s_0 = s$
  else
  generate random $x \in [0,1]$
  if $x < exp(-\delta/t)$ **then** $s_0 = s$
  $t = t * \alpha$
	Initialize population with random candidate solutions
	Evaluate each candidate
	repeat
	  repeat
	  Select parents
	  Recombine pairs of parents
	  Mutate the resulting children
	  until *iteration_count = num_mate*
	  Evaluate children
	  Select individuals for the next generation
	until Termination_Condition is satisfied

Hình 4.1: Kiến trúc hai mức của M.T.Sơn và các cộng sự [28]

Công trình của Gnanlet [23] và một công trình khác của M.T.Sơn và các cộng sự [28] đã sử dụng giải thuật mô phỏng luyện kim (*Simulated Annealing* - **SA**) để thay thế cho mức meta-level như trong phương pháp của Cortez, gọi là high-level, nhưng ở mức low-level sử dụng các biến thể về thông số của giải thuật di truyền khác nhau.

Mức high-level có nhiệm vụ tìm ra mô hình ARMA thích hợp nhất. Với mỗi mô hình tìm được ở mức high-level, mức low-level sẽ có nhiệm vụ tìm ra các tham số tốt nhất của các thành phần AR và MA trong mô hình ARMA và giải thuật GA ở mức low-level sẽ thực hiện quá trình này.

Mức high-level sử dụng một chuỗi các bit để biểu diễn mô hình ARMA, giá trị của các bit sẽ quyết định tham số của các thành phần AR và MA có được xét đến trong mô hình hay không. Lời giải s_0 (tức mô hình ARMA được khởi tạo ban đầu) được sinh ra một cách ngẫu nhiên. Và từ một mô hình đang xét, các mô hình lân cận sẽ được tìm ra thông qua một trong ba cơ chế sau:

- ***perturbation***: mỗi bit trong chuỗi bit biểu diễn mô hình ARMA (ở mức high-level) sẽ có cơ hội để chuyển trạng thái từ bit 0 lên 1 hoặc ngược lại
- ***swap***: tráo giá trị của hai vị trí trong chuỗi bit biểu diễn mô hình ARMA
- ***flip***: thực hiện đảo bit tại một vị trí nào đó

Cơ chế ***perturbation*** cho phép từ trạng thái nghiệm hiện tại chuyển sang những không gian tìm kiếm khác nhau để đảm bảo tính đa dạng (*diversity*) và làm tăng khả năng trách được nghiệm tối ưu cục bộ. Hai cơ chế còn lại đảm bảo cho giải thuật SA rà soát nghiệm trong lân cận với trạng thái hiện tại một cách cẩn thận hơn.

Đặc điểm chung của các phương pháp vừa trình bày là đều dựa trên trạng thái trạng thái sinh ngẫu nhiên cấu hình ban đầu của mô hình ARMA. Các triển khai sau đó (sử dụng phép toán lai đối với phương pháp siêu tiến hóa của Cortez hay tìm các mô hình lân cận trong giải thuật SA như vừa trình bày) đều tùy thuộc rất lớn vào mô hình. Như vậy việc tìm ra một phương pháp mở rộng không gian tìm kiếm đúng đắn là một vấn đề quan trọng có thể cải thiện được cơ chế tìm kiếm mô hình ARMA.

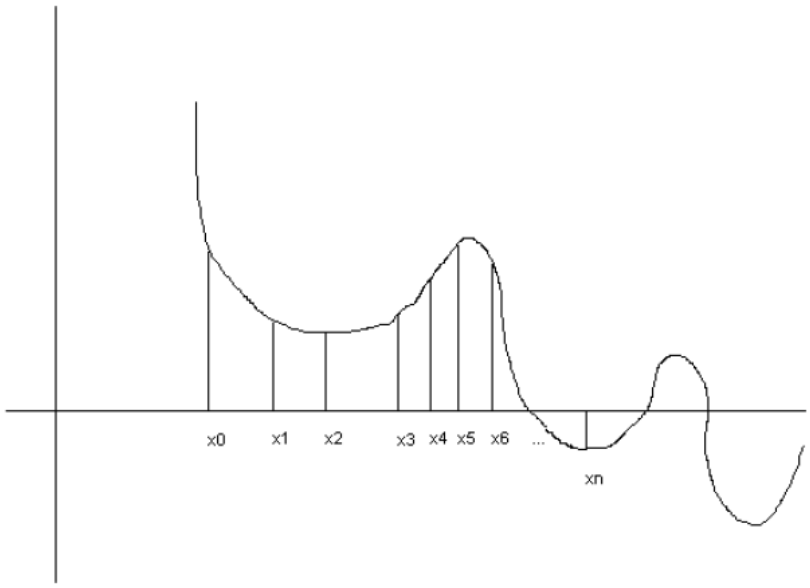
Chúng tôi đề xuất một phương pháp mở rộng không gian tìm kiếm các mô hình ARMA dựa trên giải thuật tìm kiếm Tabu. Từ giải thuật tìm kiếm Tabu chuẩn, chúng tôi sẽ

hiệu chỉnh nó để phù hợp với vấn đề xác định bậc và ước lượng các tham số của mô hình ARMA.

4.1 Giải thuật tìm kiếm Tabu

Giải thuật tìm kiếm Tabu là một kỹ thuật để giải quyết các bài toán tối ưu hóa được đề xuất bởi Glover. Chiến lược của tìm kiếm Tabu là tạo ra một danh sách giữ các lời giải đã được duyệt qua để đảm bảo rằng quá trình tìm kiếm không duyệt đến cùng một lời giải quá một lần. Danh sách này được gọi là danh sách Tabu hay danh sách cấm lưu giữ tất cả những bước chuyển (trạng thái lời giải) gần nhất trong quá trình tìm kiếm [28].

Phương pháp tìm kiếm trong không gian các lời giải là tạo ra một bước chuyển từ trạng thái lời giải đang xét x ở bước lặp thứ n đến một lời giải tốt nhất x' trong tập con $N^*(x)$ của tập hợp các lân cận của $N(x)$. x' sẽ trở thành lời giải hiện tại cho vòng lặp kế tiếp của quá trình tìm kiếm Tabu. Quá trình lặp sẽ được thực hiện cho đến khi ta đạt được lời giải x^* chấp nhận được theo một tiêu chuẩn nào đó.



Hình 4.2: Quá trình lựa chọn lời giải tốt nhất x' ở mỗi bước lặp

Khi x' không tốt hơn x, việc di chuyển trạng thái lời giải đến x' không đem lại sự cải thiện nào về chất lượng lời giải nhưng nó sẽ giúp cho quá trình tìm kiếm Tabu thoát ra khỏi tối ưu cục bộ. Vì đặc điểm không nhất thiết x' phải tốt hơn x, một cơ chế được đặt ra để loại bỏ việc lặp vòng bằng cách sử dụng danh sách Tabu để loại bỏ những lời giải được duyệt qua trước đó ra khỏi tập $N^*(x)$.

Độ dài của danh sách Tabu là một tham số của giải thuật. Chọn độ dài của danh sách Tabu như thế nào là rất quan trong, nó phải đủ dài để trách sự lặp vòng nhưng cũng phải đủ ngắn để trách việc không tìm ra được bước chuyển trạng thái là không xảy ra. Về mặt hiện thực, danh sách Tabu không nhất thiết phải được hiện thực như là một danh sách, các cấu trúc dữ liệu phức tạp hơn có thể được sử dụng để cải thiện hiệu quả việc kiểm tra trạng thái của một bước chuyển có thuộc về danh sách Tabu hay không, chẳng hạn ta có thể sử dụng cơ chế bảng băm (*hash table*) để hiện thực danh sách Tabu.

Vì bản chất của danh sách Tabu chỉ lưu giữ các đặc điểm của một lời giải chứ không phải toàn bộ chi tiết lời giải (ví dụ đối với lời giải là mô hình ARMA mà chúng tôi đề nghị khi áp dụng giải thuật tìm kiếm Tabu, đặc điểm mà chúng tôi lưu giữ lại trong danh sách Tabu để thể hiện lời giải bị cấm là các biến trễ thời gian trong từng thành phần AR và MA của mô hình ARMA), nên cần có một kỹ thuật khác để tránh việc bỏ sót lời giải tốt vì cơ chế trách lặp vòng này (có thể thấy hai lời giải cho mô hình ARMA có các đặc điểm lưu trong danh sách Tabu giống nhau nhưng đại lượng BIC đánh giá cho hai lời giải đó hoàn toàn có thể khác nhau vì giá trị ước lượng của các tham số đạt được khác nhau), và kỹ thuật mà chúng tôi muốn xem xét ở đây là **tiêu chuẩn kỳ vọng (aspiration)**.

Tiêu chuẩn kỳ vọng để cải thiện phương pháp tìm kiếm Tabu

Căn cứ vào trạng thái hiện tại của danh sách Tabu để khẳng định các bước chuyển trạng thái từ lời giải hiện có có bị cấm hay không. Vì vậy, một bước chuyển có thể bị

cấm ngay cả khi nếu áp dụng nó vào lời giải hiện tại sẽ cho ra một lời giải chưa được duyệt đến. Nói cách khác, quá trình tìm kiếm Tabu cơ bản như trình bày ở trên không những tránh việc duyệt lại những lời giải đã có từ những lần lặp trước mà còn tránh luôn cả những lời giải chỉ chia sẻ các đặc điểm giống nhau. Đây là cơ chế tránh lặp vòng, nhưng sẽ bỏ sót những lời giải tốt. Vì lý do này có một kỹ thuật gọi là **tiêu chuẩn kỳ vọng** cho phép ta ghi đè lên trạng thái bị cấm: lời giải x' đạt được từ việc thực hiện bước chuyển *m* lên trên lời giải hiện có x, ký hiệu là $x' = x \oplus \mathrm{m}$, được chấp nhận nếu bước chuyển *m* cải thiện được hàm chi phí mà không cần quan tâm đến trạng thái cấm của nó trong danh sách Tabu.

Cơ chế này tận dụng một hàm gọi là **hàm kỳ vọng *A***. Với mỗi giá trị *t* của hàm chi phí f, ***A*** tính toán chi phí mà giải thuật muốn đạt được bắt đầu từ giá trị *t*.

Tiêu chuẩn dừng cho phương pháp tìm kiếm Tabu

Quá trình lặp để dần cải thiện chất lượng lời giải sẽ được kết thúc dựa vào nhưng tiếu chí sau:

- số vòng lặp cực đại ***itermax***
- số vòng lặp liên tục nối tiếp nhau ***nitermax*** *mà* chất lượng lời giải $f(x^*)$ không được cải thiện.
- Các tham số điều khiển chính của giải thuật tìm kiếm Tabu là:
- Độ dài của danh sách Tabu
- Hàm chi phí đánh giá lời giải f
- Hàm tiêu chuẩn kỳ vọng ***A***
- Các biến xác lập điều kiện dừng *itermax* và *nitermax*
- Định nghĩa tập các lời giải lân cận $N^*(x)$ được kiểm tra tại mỗi vòng lặp

// **Bước khởi tạo**
Chọn một lời giải ban đầu x^0;
Danh sách Tabu $TL = \emptyset$;
$iter = niter = 0$;
$x^* = x^0$;
$stop = false$;
// **Bước lặp**
while *not* $stop$
    $iter = iter + 1$;
    $niter = niter + 1$;
    Xác định tập con $N^*(x) \subseteq N(x)$ gồm các phần tử
        $z = x \oplus m$ thỏa mãn
            hoặc *m* nằm ngoài TL
            hoặc $\boldsymbol{A}(z) < \boldsymbol{A}(x^*)$;
    Xác định $x' \in N^*(x)$ thỏa mãn
        $x' = argmin\{f(z)\}_{z \in N^*(x)}$;
    $x = x'$;
    if $f(z) < f(x^*)$ **then**
        $x^* = x$;
        niter = 0;
    if $iter = itermax\ or\ niter = nitermax$ **then**
        $stop = true$;
    Cập nhật danh sách Tabu TL;
// x^* là lời giải tốt nhất tìm được bởi giải thuật

Hình 4.3: Giải thuật tìm kiếm Tabu sử dụng tiêu chuẩn kỳ vọng $\boldsymbol{A}$

Trước khi trình bày tiếp giải thuật tìm kiếm Tabu được hiệu chỉnh để giải quyết vấn đề xác định bậc và ước lượng tham số của mô hình ARMA, chúng tôi sẽ giới thiệu cách thức xây dựng tập con các lời giải lân cận $N^*(x)$. Việc định nghĩa lại $N^*(x)$ chính là cốt lõi của phương pháp mở rộng không gian tìm kiếm mà chúng tôi đề nghị trong luận văn này.

Vì việc xây dựng nên tập các lời giải lân cận $N^*(x)$ sẽ phụ thuộc vào thủ tục ước lượng tham số của mô hình ARMA sử dụng giải thuật di truyền (tức mức low-level như trong các phương pháp của Cortez, Gnanlet và M.T.Sơn), nên trước hết chúng tôi sẽ trình

bày lại chi tiết cách xây dựng thủ tục sử dụng giải thuật di truyền để ước lượng tham số và chi tiết cấu hình các tham số của giải thuật, sau đây gọi là mô hình GA-ARMA

4.2 Mô hình GA-ARMA

Mô hình GA-ARMA là mô hình ARMA trong đó sử dụng giải thuật di truyền để ước lượng các tham số của mô hình.

Mỗi mô hình ARMA(p,q) sẽ được mã hóa thành một nhiễm sắc thể (NST) sử dụng trực tiếp kiểu biểu diễn thực.

$$X_t = \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}$$

ϕ_1	ϕ_2	$\cdots$	ϕ_p	θ_1	θ_2	$\cdots$	θ_q

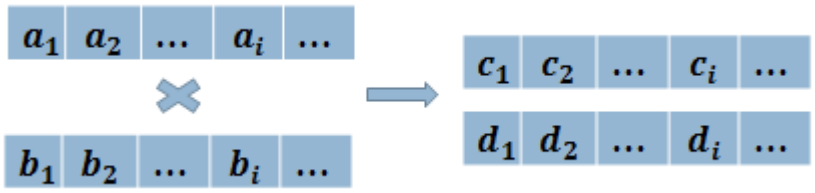
Hình 4.4: Nhiễm sắc thể biểu diễn thực đại diện trong mô hình GA-ARMA

Mỗi gen trong nhiễm sắc thể tương ứng với một tham số cần ước lượng trong mô hình. Các gen trong bước khởi tạo quần thể được sinh ngẫu nhiên trong khoảng [-1, 1]. Hàm thích nghi đánh giá chất lượng của NST được tính theo độ đo căn bậc hai của trung bình bình phương sai số (RMSE). Để tạo ra thế hệ mới, các NST cha mẹ được lựa chọn theo qui luật bánh xe Roulette.

Để làm tăng tính đa dạng của quá trình tìm kiếm, tại một thế hệ luôn đảm bảo các cá thể con sinh ra phải đôi một khác nhau, chính vì lý do này nên số lượng cá thể con sinh ra ở một thể hệ có thể bé hơn POPSIZE (tham số biểu thị số dân của giải thuật di truyền). Và như vậy các cá thể tốt nhất trong số các cá thể cha mẹ sẽ được bổ sung vào cùng với các cá thể con (sao cho đảm bảo số lượng POPSIZE) để tạo ra thế hệ kế tiếp.

4.2.1 Phép toán lai

Phép toán lai được sử dụng ở mô hình GA-ARMA là phép toán lai số học (*arithmetical crossover*) [24].



Hình 4.5: Minh họa cho phép toán lai số học

Gen ở vị trí thứ i ở các NST con được xác định theo công theo công thức sau:

$$c_i = \lambda a_i + (1 - \lambda) b_i$$

$$d_i = \lambda b_i + (1 - \lambda) a_i$$

Với λ được sinh ngẫu nhiên trong khoảng [0, 1].

4.2.2 Phép toán đột biến

Quá trình tạo đột biến được thực hiện bằng cách chọn ngẫu nhiên một trong ba phép toán tạo đột biến sau: Gaussian Perturbation [24], Relative Gaussian Perturbation và Zero-Preserving Gaussian Perturbation [25].

Gaussian Perturbation (đột biến Gaussian): phép toán đột biến được thực hiện bằng cách lấy giá trị của gen được tạo đột biến cộng thêm với một giá trị được lấy từ hàm phân bố Gaussian.

Relative Gaussian Pertubation: phép toán đột biến này được thực hiện bằng cách gây đột biến tại tất cả các gen (chứ không giới hạn tại một gen xác định như đột biến Gaussian), tại mỗi gen nhân thêm vào đại lượng (1 + g) trong đó g là giá trị được lấy từ hàm phân bố Gaussian.

Zero-Preserving Gaussian Pertubation: tương tự với phép đột biến Relative Gaussian Pertubation, tại mỗi gen cộng thêm vào đại lượng (1 – IsZero(gen))*g trong đó IsZero(x) = 1 nếu x = 0, IsZero(x) = 0 trong trường hợp khác.

4.3 Khởi tạo lời giải ban đầu đối với giải thuật tìm kiếm Tabu

Mỗi lời giải trong giải thuật tìm kiếm Tabu bao gồm các thành phần sau:

- Một chuỗi có độ dài P_{max}+Q_{max} gồm các số nhị phân {0, 1} trong đó P_{max} bit đầu tiên biểu diễn các độ trễ của thành phần AR và Q_{max} bit còn lại biểu diễn các độ trễ của thành phần MA. P_{max} và Q_{max} lần lượt là bậc tối đa của thành phần AR và MA, nhưng bậc thật sự của mô hình tùy thuộc vào giá trị của chuỗi bit.
- Một mô hình GA-ARMA được giải mã từ chuỗi bit bên trên tương tự như trong phương pháp của Cortez.

Phương pháp mở rộng không gian tìm kiếm mà chúng tôi đề xuất sẽ ảnh hưởng trực tiếp lên trên quá trình tìm tập con lân cận trong giải thuật tìm kiếm Tabu và chất lượng lời giải dựa trên tiêu chí làm cho mô hình càng ít tham số càng tốt, vì vậy lời giải khởi tạo ban đầu phải có mặt đầy đủ các độ trễ cho cả hai thành phần AR và MA. Tức là ở đây chúng tôi sẽ sử dụng loại STW đầy đủ và các biến P_{max}, Q_{max} sẽ được khởi tạo với giá trị là 13 (là giá trị được xem như là đủ để bao quát các ảnh hưởng như yếu tố mùa vụ và yếu tố xu hướng).

4.4 Phương pháp tìm tập con lân cận $\boldsymbol{N^*(x)}$

Một hiện tượng được biết đến trong lãnh vực giải thuật di truyền và lập trình tiến hóa là một vài gen có khuynh hướng biến mất trong suốt quá trình tiến hóa. Hiện tượng này được Flores và các cộng sự [26] giải thích bằng định lý của Price [27] như sau: số lượng các gen thích nghi tốt (*fit genes*) sẽ tăng qua mỗi thế hệ trong khi số lượng các gen không tốt (*unfit genes*) sẽ giảm đi.

Dựa vào hiện tượng này chúng tôi bắt đầu phân tích hành vi như vậy có xuất hiện trong quá trình học của mô hình GA-ARMA hay không. Thực nghiệm ban đầu với mô hình GA-ARMA sử dụng STW đầy đủ các độ trễ như trong 4.3 cho chúng tôi thấy rằng vài tham số (đại diện bởi các gen) ước lượng được từ mô hình ARMA bởi giải thuật di truyền có xu hướng giảm dần về 0 vào cuối quá trình tiến hóa trong khi vài tham số đóng vai trò đáng kể trong mô hình. Từ đây chúng tôi đề xuất phương pháp tạo ra các bước chuyển *m* đối với mô hình GA-ARMA để từ đó xây dựng nên tập con các lân cận $N^*(x)$ như sau:

// Mô hình GA-ARMA(P_{max} ,Q_{max})
Cửa sổ thời gian trượt đối với thành phần AR: STW_{AR} = <ar_1, ar_2, …, ar_p>, p ≤ P_{max}
Cửa sổ thời gian trượt đối với thành phần MA: STW_{MA} = <ma_1, ma_2, …, ma_q>, q ≤ Q_{max}
Trạng thái hiện tại của lời giải được biểu diễn bởi:

$$Lời\ giải\ x: X_t = \phi_{ar_1} X_{t-ar_1} + \cdots + \phi_{ar_p} X_{t-ar_p} + \theta_{ma_1} \varepsilon_{t-ma_1} + \cdots + \theta_{ma_q} \varepsilon_{t-ma_q}$$

Giải thuật di truyền trong mô hình GA-ARMA sẽ ước lượng các tham số ϕ_{ar_1}, ϕ_{ar_2}, …, ϕ_{ar_p}, θ_{ma_1}, θ_{ma_2}, …, θ_{ma_q}.

Đặt $threshold = \frac{\sum_{i=1}^{p}\left|\phi_{ar_i}\right| + \sum_{i=1}^{q}\left|\theta_{ma_i}\right|}{p+q}$

Đặt $N^*(x) = \emptyset$

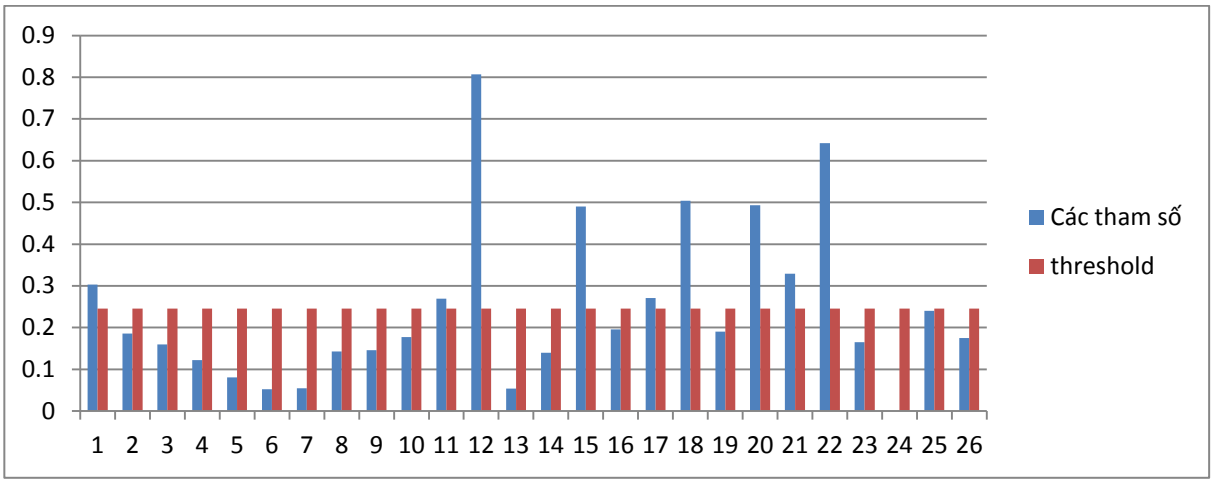
Duyệt qua các tham số ϕ_{ar_1}, ϕ_{ar_2}, …, ϕ_{ar_p}, θ_{ma_1}, θ_{ma_2}, …, θ_{ma_q}. Nếu tham số nào bé hơn $threshold$ thì:

Bước chuyển *m* được tạo ra bằng cách loại bỏ biến trễ tương ứng với tham số đó ra khỏi cửa sổ thời gian trượt STW_{AR} hoặc STW_{MA}

Kết nạp $x \oplus m$ vào $N^*(x)$

Hình 4.6: Thủ tục xác định tập con các lời giải lân cận $N^*(x)$

Mỗi bước chuyển được định nghĩa như thủ tục bên trên khi áp vào mô hình hiện tại sẽ tạo ra một mô hình mới ít hơn một tham số so với mô hình cũ. Giá trị $threshold$ để tạo ra các bước chuyển ở thủ tục trên được gọi là giá trị ngưỡng trung bình.

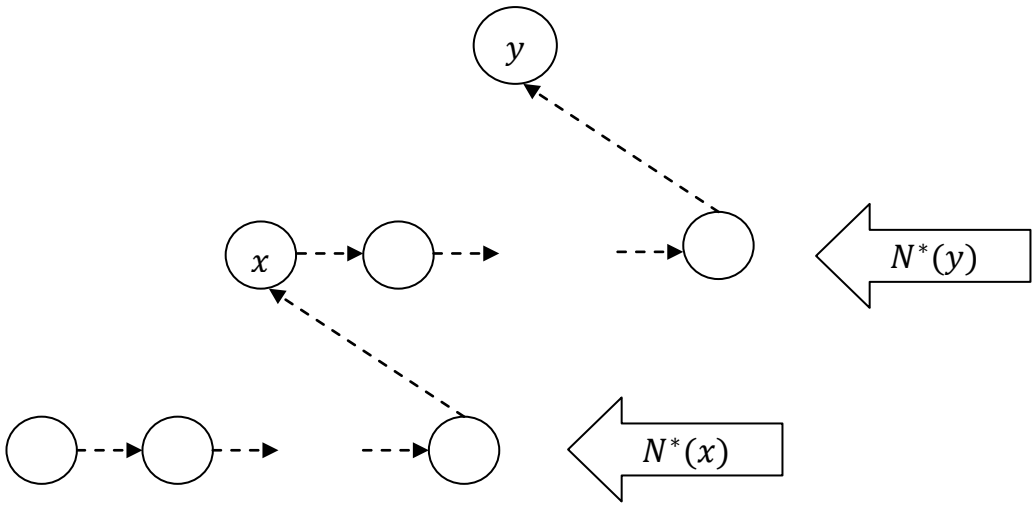


Hình 4.7: Minh họa so sánh các tham số của lời giải với giá trị ngưỡng $threshold$ để tạo ra các bước chuyển

4.5 Hiệu chỉnh giải thuật tìm kiếm Tabu

Trong giải thuật tìm kiếm Tabu, trong tập con các lân cận $N^*(x)$ chỉ có một lời giải tốt nhất được được dùng để tiếp tục quá trình mở rộng không gian tìm kiếm ở các bước lặp tiếp theo. Cơ chế này có thể bỏ qua lời giải tốt, vì vậy chúng tôi đề xuất phương pháp sau: bên cạnh việc lấy lời giải tốt nhất trong tập lân cận $N^*(x)$ như đã nói, chúng tôi cho phép giải thuật sử dụng tất cả các lời giải khác trong tập lân cận tham gia vào quá trình mở rộng không gian lời giải. Làm thế này giúp cho mô hình ARMA tốt nhất mà giải thuật hướng đến không bị bỏ qua.

Để hiện thực ý tưởng này, trước hết chúng tôi sắp xếp các lời giải trong $N^*(x)$ theo thứ tự từ tốt nhất đến xấu nhất tạo thành một danh sách liên kết lại với nhau, riêng lời giải xấu nhất trong $N^*(x)$ sẽ liên kết về chính x, vì ngay cả x (trừ trường hợp x là lời giải khởi tạo ban đầu) cũng thuộc về tập con lân cận $N^*(y)$ của một lời giải y nào đó. Cách nối vòng này để đảm bảo giải thuật sẽ quay về các lời giải ở các trạng thái trước đó để thực hiện việc chọn các lời giải còn lại trong tập lân cận thực hiện tiếp quá trình mở rộng không gian lời giải.



Hình 4.8: Kết nối các lời giải trong cùng tập con lân cận

Chúng tôi hiệu chỉnh giải thuật tìm kiếm Tabu lại như trong hình 4.9 dưới đây.

// **Bước khởi tạo**
Chọn một lời giải ban đầu x^0;
Danh sách Tabu $TL = \emptyset$;
$iter = niter = 0$;
$x^* = x^0$;
$stop = false$;
// **Bước lặp**
while *not* $stop$
    $iter = iter + 1$;
    $niter = niter + 1$;
    Xác định tập con $N^*(x) \subseteq N(x)$ gồm các phần tử (*)
        $z = x \oplus m$ thỏa mãn
            hoặc m nằm ngoài TL
            hoặc $\boldsymbol{A}(z) < \boldsymbol{A}(x^*)$;
    while $N^*(x) = \emptyset \; and \; x.next \neq x^0$ **do**
        $x = x.next$;
        Gọi lại thủ tục (*) để xác định tập $N^*(x)$;
    if $N^*(x) = \emptyset$ **then**
        $stop = true$;
        exit;
    Xác định $x' \in N^*(x)$ thỏa mãn
        $x' = argmin\{f(z)\}_{z \in N^*(x)}$;
    $x = x'$;
    if $f(z) < f(x^*)$ **then**
        $x^* = x$;
        niter = 0;
    if $iter = itermax \; or \; niter = nitermax$ **then**
        $stop = true$;
    Cập nhật danh sách Tabu TL;
// x^* là lời giải tốt nhất tìm được bởi giải thuật

Hình 4.9: Giải thuật tìm kiếm Tabu được hiệu chỉnh

Chương 5. KẾT QUẢ THỰC NGHIỆM

Trong chương này, chúng tôi trình bày các kết quả thực nghiệm từ phương pháp do chúng tôi đề xuất và tiến hành so sánh với các phương pháp meta-heuristic khác mà chúng tôi đã đề cập đến như phương pháp siêu tiến hóa của Cortez [5], phương pháp của M.T.Sơn và các cộng sự [28] cũng như các phương pháp truyền thống khác. Thống kê về thời gian chạy theo phương pháp của chúng tôi cũng được trình bày ở đây.

5.1 Dữ liệu thực nghiệm

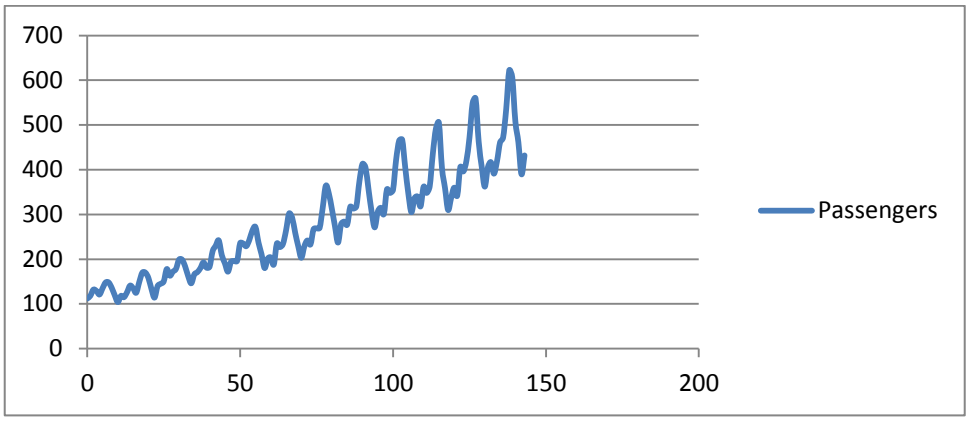
Để đánh giá kết quả, dữ liệu mà chúng tôi sử dụng trong đề tài này giống như các tập dữ liệu được sử dụng trong [5] [28]. Các tập dữ liệu này được lấy từ Time-Series Data Library (nguồn: http://robjhyndman.com/TSDL/).

Các dữ liệu này trải rộng từ dữ liệu thị trường tài chính cho đến các quá trình tự nhiên và được phân thành bốn loại *Seasonal*, *Trended*, *Seasonal and Trended* and *Nonlinear*.

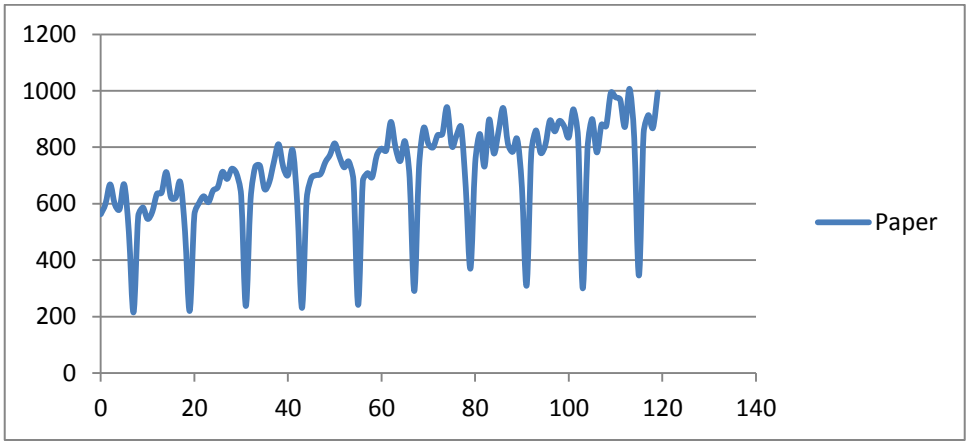
Series	Type	Domain	Description
Passengers	Seasonal	Tourism	Monthly international airline passengers
Paper	& Trended	Sales	Monthly sales of French paper
Deaths	Seasonal	Traffic	Monthly deaths & injuries in UK roads
Maxtemp		Meteorology	Maximum temperature in Melbourne
Chemical	Trended	Chemical	Chemical concentration readings
Prices		Economy	Daily *IBM* common stock closing prices
Sunspots	Nonlinear	Physics	Annual Wolf's Sunspot Numbers
Kobe		Geology	Seismograph of the Kobe earthquake

Bảng 5.1: Phân loại các tập dữ liệu được sử dụng để thực nghiệm

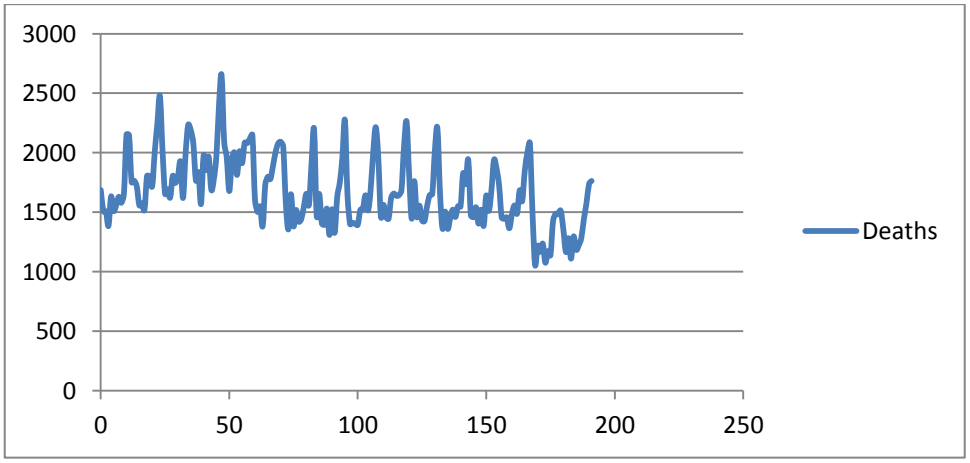
Các hình biểu diễn đồ thị của các chuỗi dữ liệu thời gian:



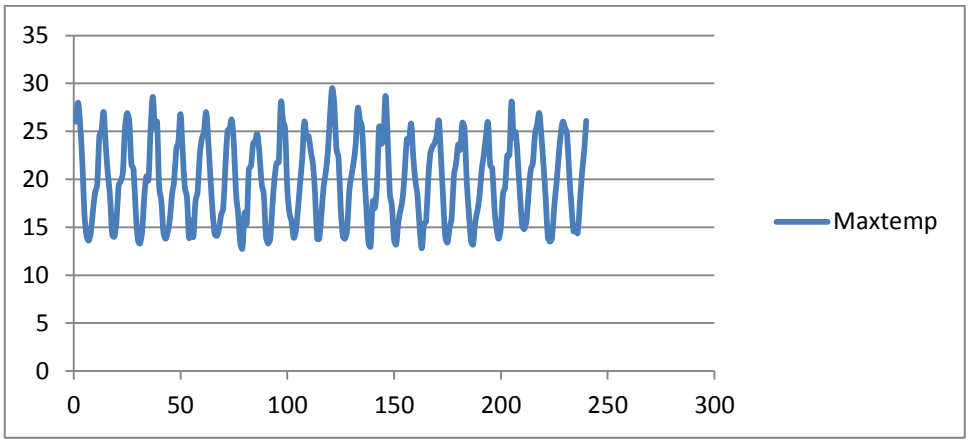
Hình 5.1: Đồ thị chuỗi dữ liệu Passengers



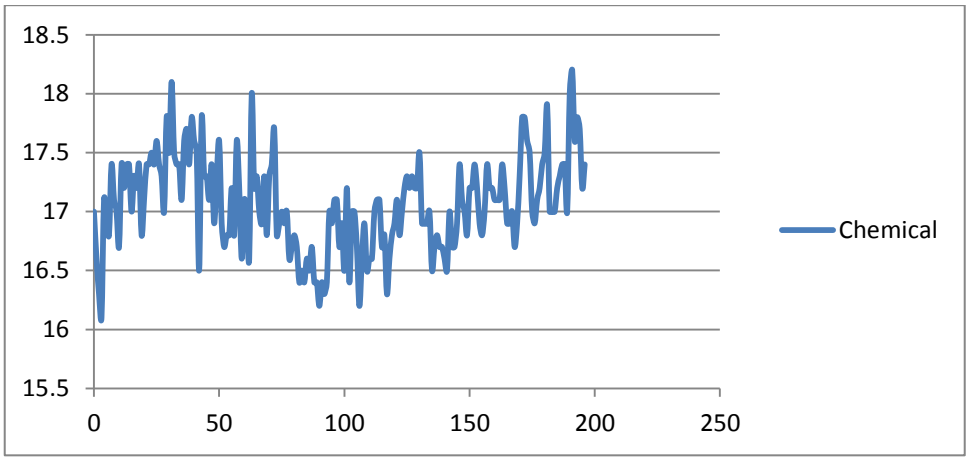
Hình 5.2: Đồ thị chuỗi dữ liệu Paper



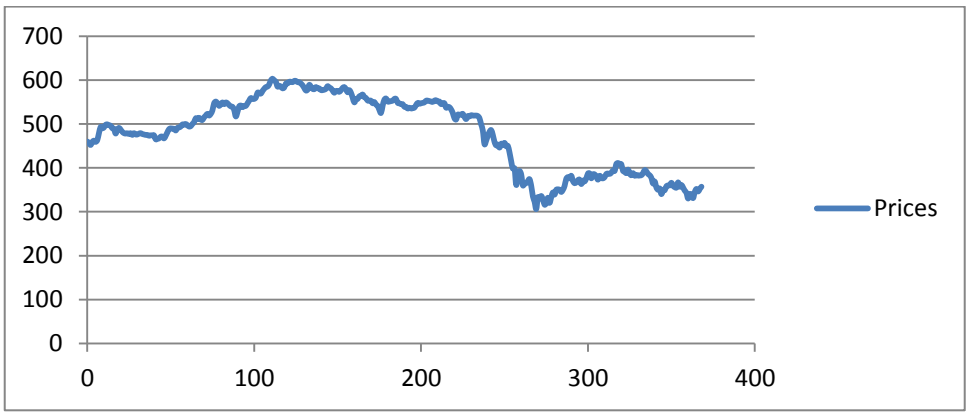
Hình 5.3: Đồ thị chuỗi dữ liệu Deaths



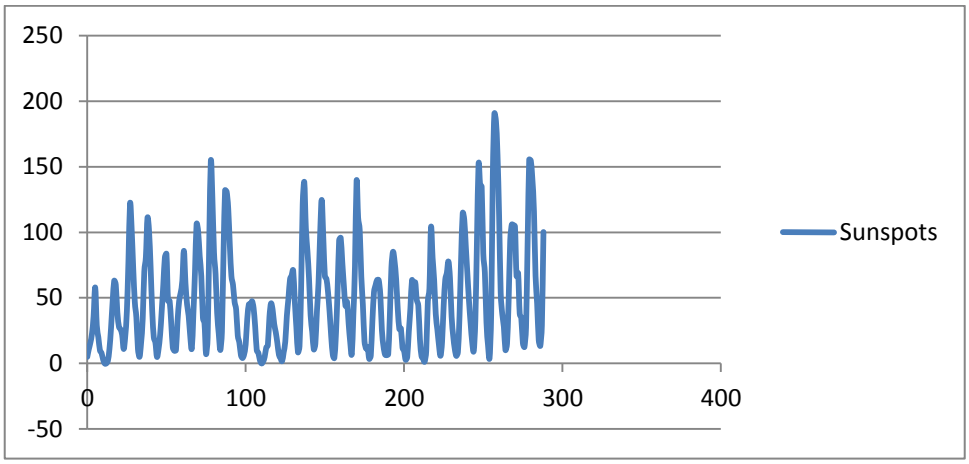
Hình 5.4: Đồ thị chuỗi dữ liệu Maxtemp



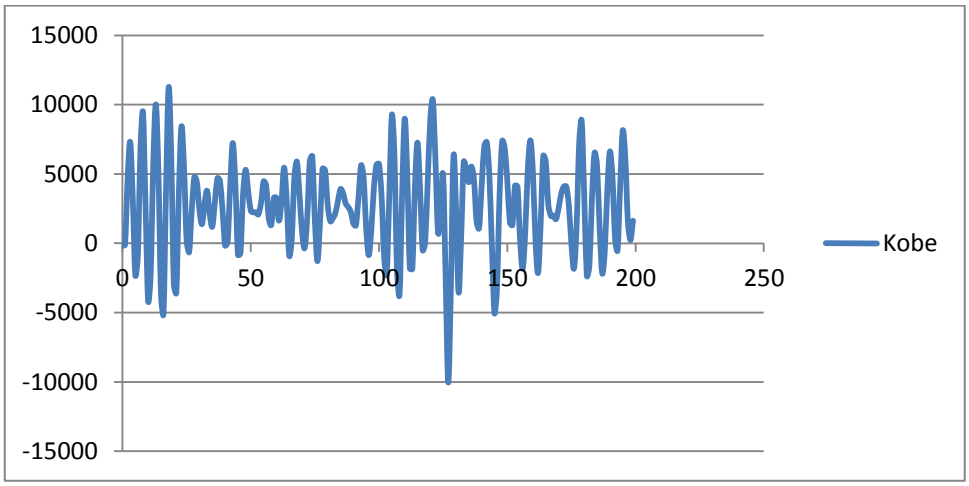
Hình 5.5: Đồ thị chuỗi dữ liệu Chemical



Hình 5.6: Đồ thị chuỗi dữ liệu Prices



Hình 5.7: Đồ thị chuỗi dữ liệu Sunspots



Hình 5.8: Đồ thị chuỗi dữ liệu Kobe

5.2 Kết quả thực nghiệm và đánh giá

Chương trình được thực hiện bằng ngôn ngữ Java SE 1.6, chạy trên máy Intel Core 2, Duo, CPU 2.33Ghz, RAM 8GB.

Giải thuật di truyền dùng để hiện thực mô hình GA-ARMA được chạy với các tham số sau:

- Kích thước quần thể là 50
- Giải thuật di truyền sẽ dừng sau 1000 thế hệ

- Xác suất lai $\mathbf{P_c}$ là 0.8: 80% số cá thể của quần thể mới là do phép toán lai tạo ra, 20% số cá thể trong quần thể mới được giữ lại từ thế hệ trước đó.
- Xác suất tạo đột biến $\mathbf{P_m}$ là 0.3 cho các cá thể trong quần thể mới.
- Giải thuật tìm kiếm Tabu hiệu chỉnh được chạy với các tham số sau:
- Số vòng lặp cực đại ***itermax* = 500**
- Số vòng lặp liên tục nối tiếp nhau mà chất lượng lời giải $f(x^*)$ không được cải thiện ***nitermax*** được chúng tôi sử dụng với một giá trị cực lớn để điều kiện dừng của chương trình không phụ thuộc vào tham số này vì chúng tôi không xét ràng buộc về mặt thời gian mà mục tiêu hướng đến tìm được lời giải tốt nhất có thể.
- Độ dài của danh sách Tabu: hiện thực với cơ chế hashing trong Java, chúng tôi không xét ràng buộc về không gian bộ nhớ, độ dài danh sách Tabu mà chúng tôi sử dụng là lớn nhất có thể.
- Hàm chi phí đánh giá lời giải f là đại lượng RMSE
- Hàm tiêu chuẩn *aspiration* $\boldsymbol{A}$ mà chúng tôi sử dụng là hàm hợp sử dụng cả hai đại lượng RMSE và BIC, nghĩa là nếu lời giải lân cận thỏa mãn tiêu chuẩn hoặc đại lượng RMSE được cải thiện hoặc đại lượng BIC được cải thiện so với giá trị hiện tại của lời giải đang xét thì được xem như thỏa mãn tiêu chuẩn *aspiration*.
- Định nghĩa tập các lời giải lân cận $N^*(x)$ giống thủ tục trình bày trong hình 4.6
- Khả năng dự báo của mô hình ARMA tìm được bởi phương pháp do chúng tôi đề nghị được đánh giá bằng hai độ đo sau:
- Căn bậc hai trung bình bình phương lỗi RMSE
- Chuẩn hóa trung bình bình phương lỗi NMSE (Normalized Mean Squared Error)

$$NMSE = \frac{SSE}{\sum_{i=1}^{l}(x_i - \bar{x})^2}$$

Trong đó là l là số giá trị dùng để kiểm tra và $\bar{x}$ là giá trị trung bình của chuỗi dữ liệu. Chi tiết về các đại lượng *SSE* và *RMSE* đã được trình bày trong chương 3.

Mỗi tập dữ liệu sử dụng để đánh giá thực nghiệm được chúng tôi chia thành tập huấn luyện và tập kiểm thử. Tập huấn luyện lấy ra từ 90% số phần tử đầu tiên của tập dữ liệu ban đầu và tập kiểm thử lấy 10% số phần tử còn lại trong tập dữ liệu ban đầu. Chúng tôi xác định bậc và ước lương tham số của mô hình ARMA thể hiện qua việc chạy giải thuật tìm kiếm Tabu hiệu chỉnh chỉ sử dụng đến tập huấn luyện. Tập kiểm thử được sử dụng để so sánh khả năng dự báo của phương pháp do chúng tôi đề nghị và các phương pháp khác.

Các mô hình ARMA tốt nhất đạt được từ phương pháp của chúng tôi để trình bày sau đây trong bảng **5.2**. Với mỗi chuỗi dữ liệu, chúng tôi liệt kê các biến trễ trong cửa sổ thời gian trượt của các thành phần AR và MA cũng như tổng số lượng các tham số của cả hai thành phần AR và MA.

Series	**AR**	**MA**	**p**
Passengers	12	1, 2, 3, 4, 5, 12, 13	8
Paper	12	2, 6, 10, 12	5
Deaths	1, 2, 12, 13	2, 12	6
Maxtemp	1, 3, 11, 12	3, 12	6
Chemical	1, 3, 4, 7, 9	1, 13	7
Prices	1	5, 6, 9, 11, 13	6
Sunspots	1, 2, 7, 10	2, 9, 11	7
Kobe	1, 2, 3, 6	1, 3, 6, 9, 11	9

Bảng 5.2: Những mô hình ARMA tốt nhất tìm được bởi phương pháp đề nghị

Bảng 5.3 đưa ra kết quả so sánh phương pháp của chúng tôi đề nghị, gọi tắt là Tabu-SA, với phương pháp của M.T.Sơn và các cộng sự, gọi tắt là SAGA [28], phương pháp siêu tiến hóa của Cortez và các phương pháp truyền thống được sử dụng để so sánh với

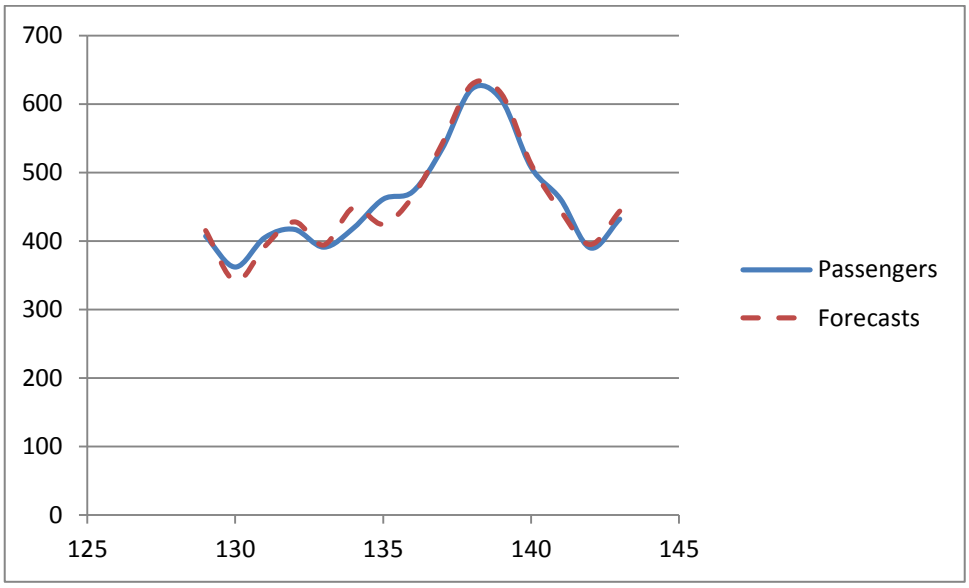
phương pháp siêu tiến hóa trong [5]. Các giá trị được in ra trong bảng ứng với từng phương pháp là đại lượng RMSE trên tập kiểm thử (trong ngoặc là giá trị của đại lượng NMSE). Kết quả thực nghiệm cho thấy phương pháp đề nghị của chúng tôi có cải thiện kết quả trên nhiều chuỗi dữ liệu nếu so với các phương pháp SAGA và Meta-GAs.

Series	ES	ARIMA	Meta-GAs	SAGA	Tabu-GA
Passengers	16.5 (0.7%)	17.8 (0.81%)	17.2 (0.75%)	17.74 (0.83%)	**15.68 (0.65%)**
Paper	49.2 (4.4%)	61.0 (6.8%)	52.5 (5%)	49.17 (4.39%)	**41.57 (3.14%)**
Deaths	**135 (37%)**	144 (42%)	137 (38%)	142 (41%)	138.77 (13.56%)
Maxtemp	**0.72 (2.5%)**	1.07 (5.6%)	0.93 (4.3%)	0.85 (3.6%)	0.80 (3.17%)
Chemical	0.35 (51%)	0.36 (53%)	0.34 (48%)	**0.33 (44.89%)**	0.33 (45.93%)
Prices	7.54 (0.39%)	7.72 (0.41%)	**7.48 (0.38%)**	7.54 (0.39%)	7.63 (0.39%)
Sunspots	28.4 (35%)	21.4 (20%)	17.6 (14%)	**16.57 (12%)**	18.11 (14.44%)
Kobe	3199 (105%)	582 (3.5%)	492 (2.5%)	491 (2.5%)	**395.26** (1.61%)

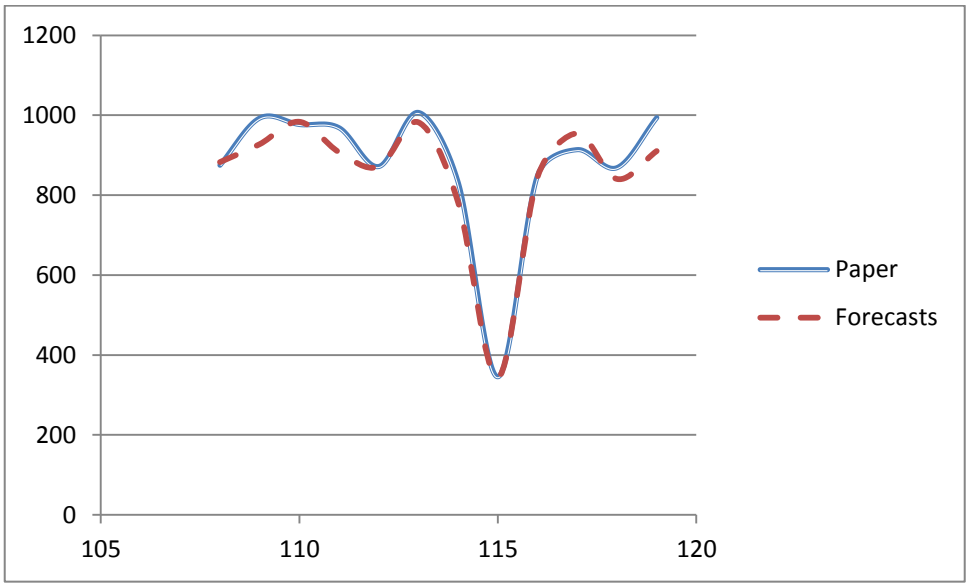
Bảng 5.3: So sánh kết quả của các phương pháp dự báo khác nhau

Phương pháp Tabu-GA đều cải thiện chất lượng lời giải đối với 5/8 chuỗi dữ liệu nếu so với phương pháp Meta-GA và cả phương pháp SAGA. Đối với chuỗi dữ liệu có yếu tố mùa (*Seasonal*), phương pháp làm trơn hàm mũ (ES) vẫn cho kết quả tốt nhất vì đây là phương pháp được thiết kế riêng cho những chuỗi dữ liệu dạng này.

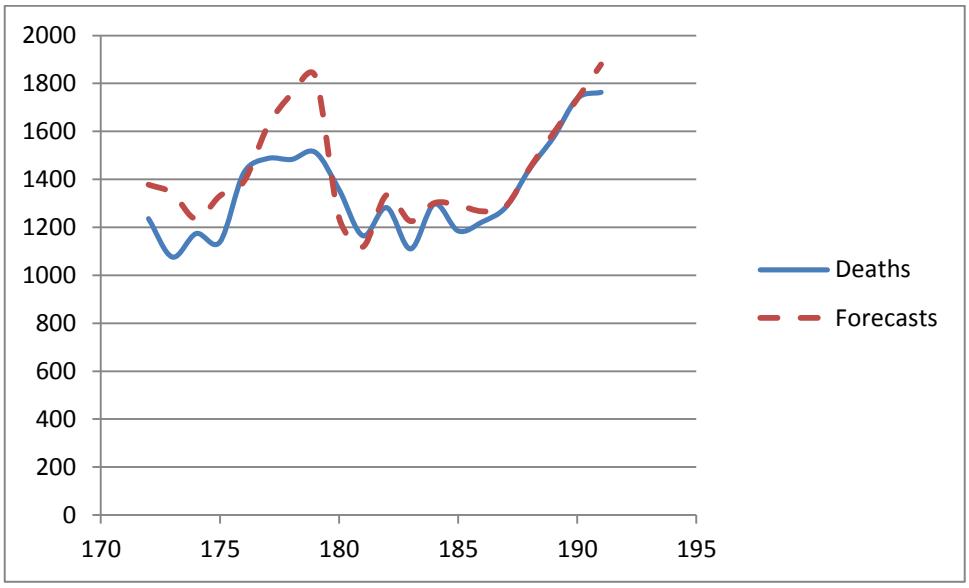
Các hình sau biểu diễn đồ thị dự báo giá trị của 10% các phần tử cuỗi của các chuỗi dữ liệu được dùng cho thực nghiệm:



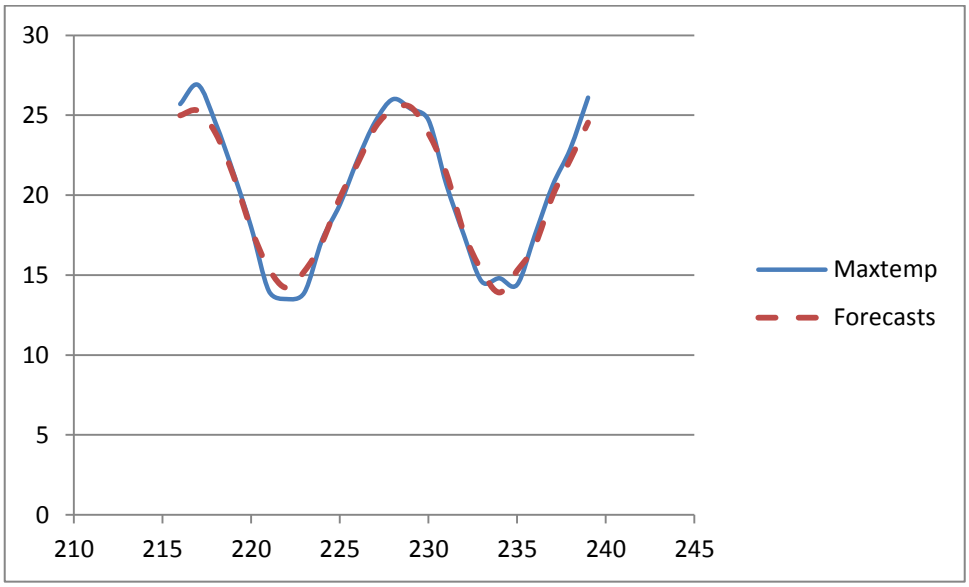
Hình 5.9: Đồ thị dự báo tập dữ liệu Passengers



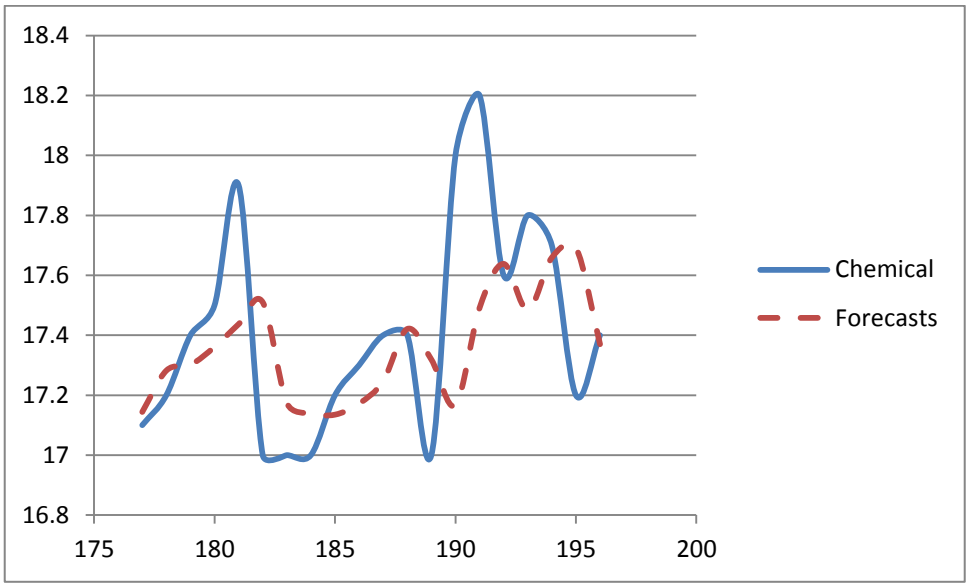
Hình 5.10: Đồ thị dự báo tập dữ liệu Paper



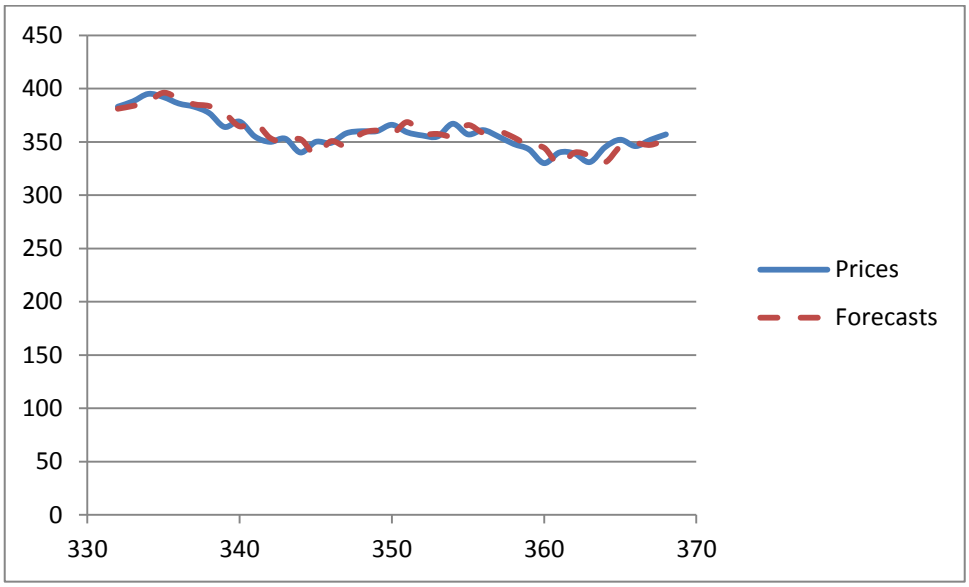
Hình 5.11: Đồ thị dự báo tập dữ liệu Deaths



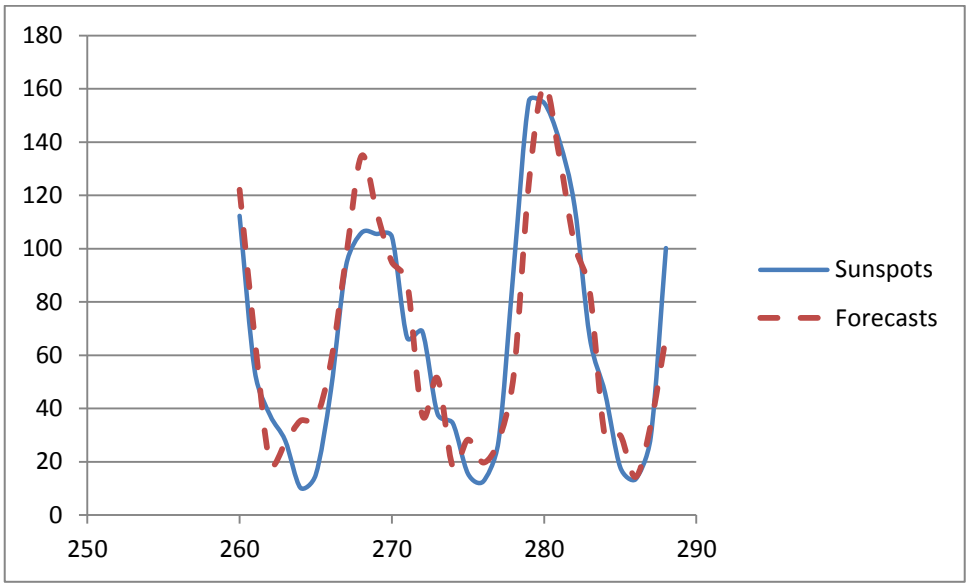
Hình 5.12: Đồ thị dự báo tập dữ liệu Maxtemp



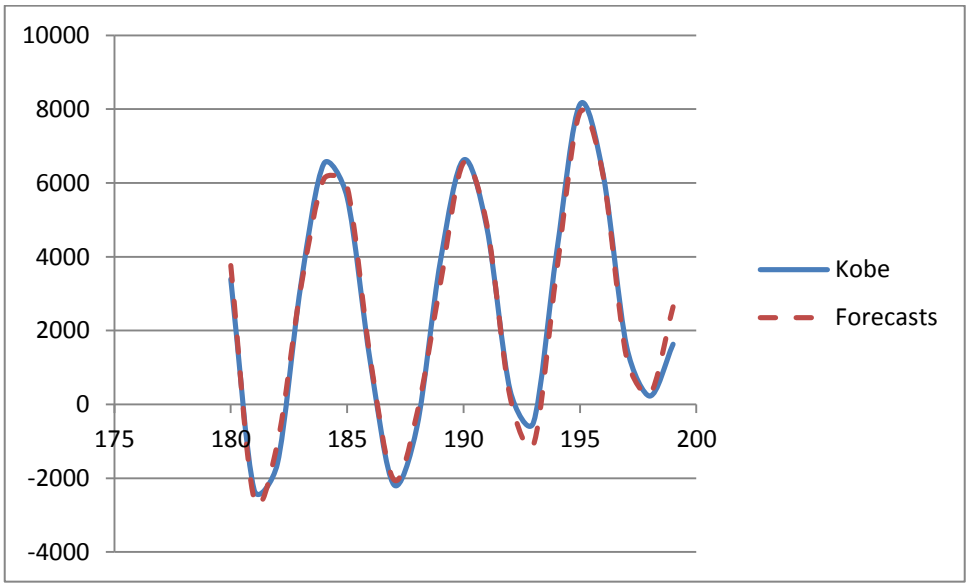
Hình 5.13: Đồ thị dự báo tập dữ liệu Chemical



Hình 5.14: Đồ thị dự báo tập dữ liệu Prices



Hình 5.15: Đồ thị dự báo tập dữ liệu Sunspot



Hình 5.16: Đồ thị dự báo tập dữ liệu Kobe

Thời gian chạy tương ứng với số vòng lặp ***itermax*** = 500 tương ứng với các chuỗi dữ liệu dùng trong thực nghiệm được liệt kê trong bảng 5.4 dưới đây. Thời gian này có thể

xem như là thời gian ước lượng cực đại của giải thuật ứng với điều kiện dừng ***itermax*** = 500 của giải thuật Tabu hiệu chỉnh trong phương pháp của chúng tôi. Sở dĩ chúng tôi gọi đây là thời gian ước lượng cực đại vì trên thực tế, ta có thể sử dụng thêm tham số ***nitermax*** của giải thuật để cho phép giải thuật kết thúc sớm hơn nếu như sau một số lần các vòng lặp (được chỉ định bởi ***nitermax***) mà chất lượng của lời giải không được cải thiện thêm.

Series	Data Points	Thời gian chạy Tabu-SA
Passengers	144	3 giờ 33 phút
Paper	120	2 giờ 46 phút
Deaths	192	2 giờ 51 phút
Maxtemp	240	2 giờ 56 phút
Chemical	197	2 giờ 4 phút
Prices	369	3 giờ 42 phút
Sunspots	289	2 giờ 38 phút
Kobe	200	3 giờ 44 phút

Bảng 5.4: Thời gian chạy giải thuật Tabu-SA
của các chuỗi dữ liệu thực nghiệm

Bảng thời gian chạy của giải thuật Tabu-SA cho thấy phương pháp mà chúng tôi đề nghị trong luận văn này có thể chấp nhận được về mặt thời gian.

Chương 6. KẾT LUẬN

Trong chương này, chúng tôi tổng kết lại những việc đã làm được và đề xuất các hướng mở rộng để có thể phát triển đề tài.

6.1 Tổng kết

Trong đề tài này chúng tôi đã thực hiện nghiên cứu bài toán dự báo chuỗi thời gian sử dụng các phương pháp meta-heuristic để xác định bậc và ước lượng các hệ số của mô hình ARMA. Chúng tôi đã tìm hiểu cách thức kết hợp giữa giải thuật di truyền và giải thuật di truyền, giải thuật mô phỏng luyện thép với giải thuật di truyền trong các meta-heuristic sử dụng kiến trúc hai mức, mỗi mức sử dụng một giải thuật tìm kiếm cục bộ riêng và kết hợp hai mức lại với nhau theo cách kết hợp các giải thuật vừa nêu ra.

Từ những nghiên cứu này, chúng tôi đã đề xuất một phương pháp kết hợp khác giữa giải thuật tìm kiếm Tabu và giải thuật di truyền, trong đó giải thuật Tabu đảm nhận việc xác định bậc của mô hình ARMA và giải thuật di truyền đảm nhận việc ước lượng các hệ số của mô hình. Giải thuật tìm kiếm Tabu đã được chúng tôi hiệu chỉnh để phù hợp với việc xác định bậc của mô hình ARMA, trong phương pháp đề nghị của chúng tôi cách thức mở rộng không gian tìm kiếm các mô hình ARMA trong bước xác định các lời giải lân cận của giải thuật tìm kiếm Tabu là quan trọng, nó dựa trên việc đánh giá những gen nào trong NST biểu diễn cho mô hình ARMA là gen yếu cần phải được loại bỏ, từ đó cho thấy việc thực hiện các bước chuyển trong giải thuật Tabu mà chúng tôi đề xuất là có định hướng chứ không mang tính ngẫu nhiên như các phương pháp meta-heuristic trước đó. Giải thuật di truyền trong phương pháp giải quyết vấn đề của chúng tôi sử dụng các biến thể khác nhau của các phép toán lai ghép và phép toán đột biến dựa trên những tổng kết gần đây về các heuristic được áp dụng trong các bài toán tối ưu số (*numerical optimization*) (ước lượng các hệ số của mô hình ARMA là bài toán thuộc dạng này). Kết quả thực nghiệm đã chứng minh được hiệu quả của phương pháp giải quyết mà chúng tôi thực hiện trong luận văn này.

6.2 Hướng phát triển đề tài

Đề tài đã đưa ra một giải pháp mới để xác định bậc và ước lượng các hệ số của mô hình ARMA trong bài toán dự báo chuỗi thời gian. Tuy nhiên để cải thiện hướng nghiên cứu này chúng ta cần bổ xung các tiếp cận nghiên cứu mới trong tương lai như sau:

- Với kết quả thực nghiệm như đã trình bày trong chương 5, vẫn có 3/8 chuỗi dữ liệu thực nghiệm mà phương pháp đề xuất của chúng tôi vẫn không cải thiện nhiều so với các phương pháp khác. Một điều dễ nhận thấy là dữ liệu thực nghiệm không qua giai đoạn tiền xử lý dữ liệu nào trước khi chạy trực tiếp với giải thuật. Do giới hạn về thời gian nghiên cứu, nên chúng tôi chưa thể áp dụng các kiểu tiền xử lý dữ liệu như trong phương pháp của Box-Jenkins (chẳng hạn như phương pháp chuyển dữ liệu sang dạng logarithm). Dĩ nhiên qui trình tiền xử lý dữ liệu này cũng hợp nhất với phương pháp xác định bậc và ước lượng tham số của mô hình một cách tự động chứ không dựa vào năng lực cũng như kinh nghiệm của người làm dự báo theo phương pháp Box-Jenkins.
- Mặc dù thời gian chạy của giải thuật trong phương pháp do chúng tôi đề xuất dừng ở mức có thể chấp nhận được, nhưng việc cải thiện tốc độ cũng là một trong những vấn đề quan trọng cần xem xét đến nhưng một trong số các hướng phát triển tiếp đề tài này.

TÀI LIỆU THAM KHẢO

[1] A.S.Weigend and N.A.Gershenfeld, editors. **Time Series Prediction: Forecasting the Future and Understanding the Past**. *Addison Wesley, 1993*.

[2] G.E.P.Box, G.M.Jenkins and G.C.Reinsel. **Time Series Analysis: Forecasting and Control**, *San Francisco: Holden-Day, 1994*.

[3] Chris Chatfield. **Time-Series Forecasting**. *Chapman and Hall/CRC, 2000*.

[4] P.Cortez, M.Rocha and J.Neves. **A Meta-Genetic Algorithm for Time Series Forecasting**. In Luís Torgo (Ed.), *Proceedings of Workshop on Artificial Intelligence Techniques for Financial Time Series Analysis (AIFTSA -01), 10th Portuguese Conference on Artificial Intelligence (EPIA'01), pp. 21-31, Porto, Portugal, December, 2001*.

[5] P. Cortez, M. Rocha and J. Neves. **Genetic and Evolutionary Algorithms for Time Series Forecasting**. *In L. Monostori, J. Váncza and M. Ali (Eds), Lecture Notes in Artificial Intelligence 2070, pp. 393-402, Budapest, Hungary, June, 2001. Springer, ISBN:3-540-42219-6.*

[6] A.M.Fraser and A.Dimitriadis**. Forecasting Probability Densities by Using Hidden Markov Models with Mixed States**. 1993.

[7] C.C.Holt, **Forecasting Seasonals and Trends by Exponentially Weighted Moving Averages**, *unpublished research report, Carnegie Institute of Technology, Pittsburgh, 1957.*

[8] E.J.Kostelich and D.P.Lathrop. **Time Series Prediction by Using the Method of Analogues**. *1993*.

[9] J.G.Gooijer and R.J.Hyndman. **25 years of time series forecasting**, *International Journal of Forecasting 22 (2006) 443– 473*.

[10] J.Han and M.Kamber, **Data Mining: Concepts and Techniques, Second Edition**, *2006*.

[11] A. Lapedes and R. Farber. **Nonlinear signal processing using neural networks**. *Technical Report LA-UR-87-2662, Los Alamos National Laboratory, Los Alamos, NM, 1987*.

[12] K. Lang and G. Hilton. **A time-delay neural network architecture for speech recognition**. *Technical Report CMU-CS-88-152, Carnegie Mellon University, Pittsburgh, PA, 1988*.

[13] M.C. Mozer. **Neural Network Architectures for Temporal Sequence Processing**, *pages 243-264. Addison Wesley, 1993*.

[14] Minerva et al., **Building ARMA Models with Genetic Algorithms**, *2001, E.J.W. Boers et al. (Eds.) EvoWorkshop , LNCS 2037, pp. 335-342.*

[15] A.Waibel. **Modular construction of time-delay neural networks for speech recognition**. *Neur. Comp., 1(1):39-46, 1989.*

[16] R.S.Pindyck and D.L.Rubinfeld. **Econometric Models and Economic Forecasts**, *Third Edition, McGraw-Hill, 1991*.

[17] E.A.Wan. **Time Series Prediction by Using a Connectionist Network with Internal Delay Line**, *pages 195-217. Addison Wesley, 1993*.

[18] P.Werbos. Beyond Regression: **New Tools for Prediction and Analysis in the Behavioural Sciences**. *PhD thesis, Harvard University, Cambridge, MA, 1974*.

[19] P. Werbos. **Generalization of backpropagation with application to a recurrent gas market model**. *Neur. Net., 1:339-356, 1988.*

[20] A.S.Weigend, B.A.Huberman, and D. E. Rumelhart. **Predicting the future: A connectionist approach**. *International Journal of Neural Systems, 1:193-209, 1990.*

[21] J.H.Holland. **Adaptation in natural and artificial Systems**. *The University of Michigan Press, Ann Arbor, Michigan, 1975.*

[22] G.Schwarz. **Estimating the Dimension of a Model**. *Annals of Statistics, 6:461-4, 1978.*

[23] Adelina Gnanlet and Chandrasekharan Rajendran. **Meta-Heuristics in ARMA Forecasting**. *CJOM, Vol. 7(1), 2009, 38-48.*

[24] Z. Michalewicz. **Genetic Algorithms + Data Structures = Evolution Programs**. *Springer-Verlag, USA, third edition, 1996.*

[25] Daniel A. Spieleman and Shang-Hua Teng. **Smoothed Analyis of Algorithms and Heuristics: Progress and Open Questions**. *Foundation of Computational Mathematics, Santander 2005. Cambride University Press (2006).*

[26] Juan J. Flores, Héctor Rodríguez, and Mario Graff. **Reducing the search space in evolutive design of ARIMA and ANN models for time series prediction**. *Proceeding MICAI'10 Proceedings of the 9th Mexican international conference on Artificial intelligence conference on Advances in soft computing: Part II Pages 325-336, volume 6438 of Lecture Notes in Computer Science. Springer, 2010.*

[27] Price G. R., **Selection and covariance**. *Nature 227, 520-521 (1970).*

[28] Mai Thai Son et al. **A New Approach to Time Series Forecasting using Simulated Annealing Algorithms**. *ACOMP 2010*.

[29] Dương Tuấn Anh. **Bài giảng môn học "Lập trình Logic và ràng buộc".** *Khoa Khoa học và kỹ thuật máy tính, ĐHBKTPHCM, 2012*.

[30] Nguyễn Xuân Hùng. **Sử dụng giải thuật di truyền tinh chỉnh cấu hình mạng nơron cho công tác dự báo dữ liệu chuỗi thời gian.** *Luận văn tốt nghiệp đại học, 2011, Khoa học và kỹ thuật máy tính, ĐHBKTPHCM.*

LÝ LỊCH TRÍCH NGANG

Họ và tên: Lâm Hoàng Vũ

Ngày sinh: 14/10/1981

Nơi sinh: Quảng Ngãi

Địa chỉ liên lạc: 100 Trần Văn Dư, phường 13, Quận Tân Bình, TP. Hồ Chí Minh

Email: lamhoangvu@gmail.com

QUÁ TRÌNH ĐÀO TẠO

Thời gian	Trường đào tạo	Chuyên ngành	Trình độ
1999 - 2005	Trường Đại Học Bách Khoa – Đại Học Quốc TP. Hồ Chí Minh	Điện – Điện Tử	Kỹ sư
2008 - 2010	Trường Đại Học Bách Khoa – Đại Học Quốc TP. Hồ Chí Minh	Khoa Học Máy Tính	Thạc sĩ

QUÁ TRÌNH CÔNG TÁC

Thời gian	Đơn vị công tác	Vị trí công tác
2004 - 2009	Công ty Silicon Design Solution, 3E/15 Phổ Quang, phường 4, Quận Tân Bình, TPHCM.	Lập trình viên
2009 - nay	Công ty eSilicon Vietnam, Lầu 9, tòa nhà eTown, 364 Cộng Hòa, phường 13, Quận Tân Bình, TPHCM.	Lập trình viên