AI Application Programming
by M. Tim Jones
ISBN:1584502789
Charles River Media © 2003
The purpose of this book is to demystify the techniques associated with the field of artificial intelligence. It covers both the theory and the practical applications to teach developers how to apply AI techniques in their own designs.

다트머스 학술회의 튜링테스트 맥컬럭·피트 신경망	탐색 문제풀이 추론 계획 정리증명	전문가 시스템 오류 역전파 학습	인터넷, WWW IBM Deep Blue	딥 페이스 알파Go 딥 페이크 고양이 학습 딥러닝 IBM 왓슨 구글 다국어 번역
1943 1950 1956		1986	1997	2000 2010 2011 2012 2014 2016
1043~1950s	1960~1970s	1980s	1990s	2000s~
인공지능 탄생 희망의 시작	1st 겨울 좌절, 조용한 전진 계산기능 한계 논리체계 한계	인공지능 붐 전문가 시스템 성공	2nd 겨울 의심, 빠른 전진 데이터 부족	끝 모를 전진 빅 데이터, 딥러닝 CPU 처리속도 초고속 향상, GPU, TPU 메모리 등 하드웨어 비용 감소

AI Application Programming
by M. Tim Jones
ISBN:1584502789
Charles River Media © 2003
The purpose of this book is to demystify the techniques associated with the field of artificial intelligence. It covers both the theory and the practical applications to teach developers how to apply AI techniques in their own designs.

Table of Contents

Chapter 1: History of AI

In this initial chapter, we'll begin with a short discussion of artificial intelligence (AI) and a brief history of modern AI. Some of the more prominent researchers will also be discussed, identifying their contributions to the field. Finally, the structure of the book is provided at the end of this chapter, identifying the methods and techniques detailed within this text.

What is AI?

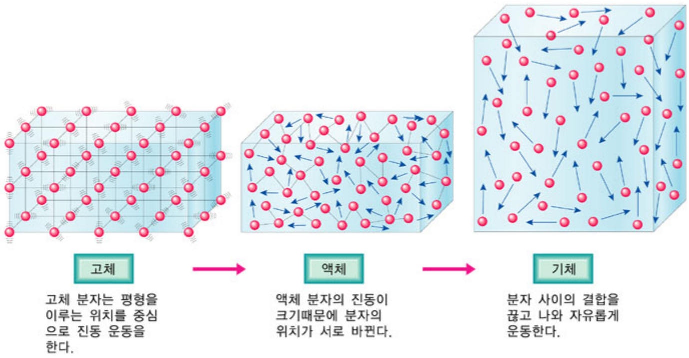
Artificial intelligence is the process of creating machines that can act in a manner that could be considered by humans to be intelligent. This could be exhibiting human characteristics, or much simpler behaviors such as the ability to survive in dynamic environments.

Chapter 2: Simulated Annealing

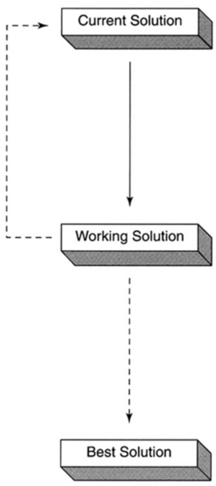
- Download CD Content

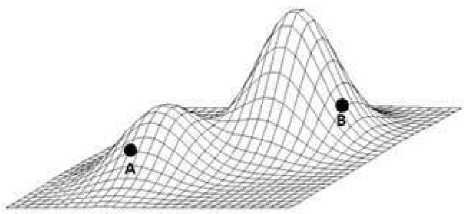
In this chapter, we'll investigate an optimization method called simulated annealing. As the name implies, the search method mimics the process of annealing. Annealing is the physical process of heating and then cooling a substance in a controlled manner. The desired result is a strong crystalline structure, compared to fast untempered cooling which results in a brittle defective structure. The structure in question is our encoded solution, and the temperature is used to determine how and when new solutions are accepted.

2장 모의 담금질

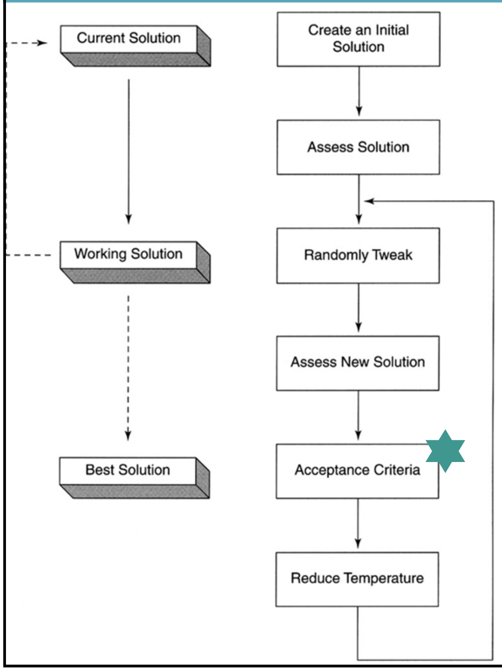


2장 모의 담금질, 알고리즘





2장 모의 담금질, 알고리즘



초기 soln(답)

soln 평가

soln 변형

new soln 평가

교체 여부 평가

온도 조절

2장 모의 담금질, 예제: N-Queen 문제

For this algorithm, we'll look at a very famous problem that has been attacked by a wide variety of search algorithms. The N-Queens problem (or NQP) is defined as the placement of N queens on an N-by-N board such that no queen threatens any other queen using the standard rules of chess (see Figure 2.3).

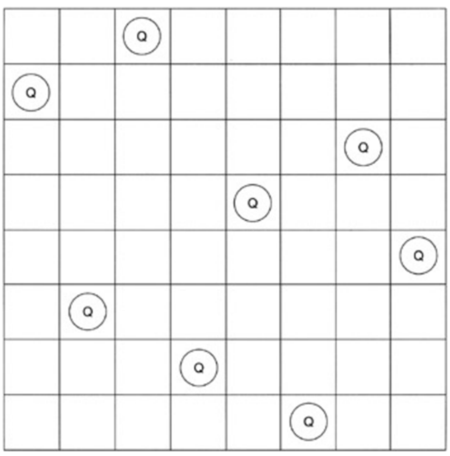


Figure 2.3: One of 92 8-Queens solutions.

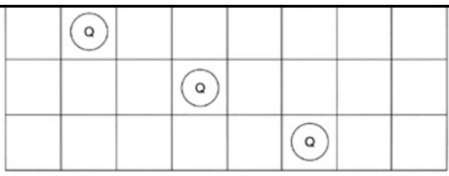


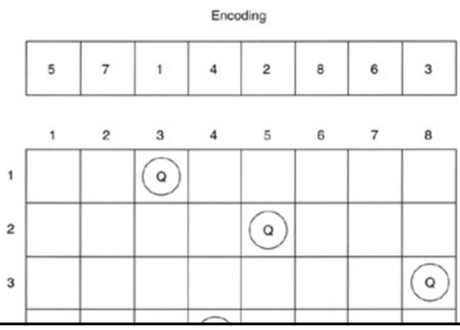
Figure 2.3: One of 92 8-Queens solutions.

The 8-Queens problem was first solved in 1850 by Carl Friedrich Gauß. The search algorithm, as can be inferred by the date of the solution, was trial and error. The N-Queens problem has since been solved using depth-first search (1987), divide and conquer (1989), genetic algorithms (1992), and a variety of other methods. In 1990, Rok Sosic and Jun Gu solved the 3,000,000-Queens problem using local search and conflict minimization [Schaller 2001].

Solution Encoding

The encoding for the N-Queens solution is a standard one that takes into account the final solution, and thus reduces the search space. Note from Figure 2.3 that only one queen can be found in each row and in each column. This constraint makes it much easier to create an encoding that will be manipulated by the simulated annealing algorithm.

Since each column contains only one queen, an N-element array will be used to represent the solution (see Figure 2.4).



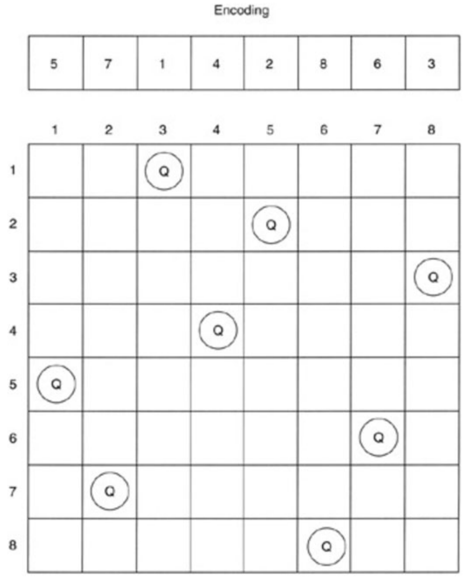
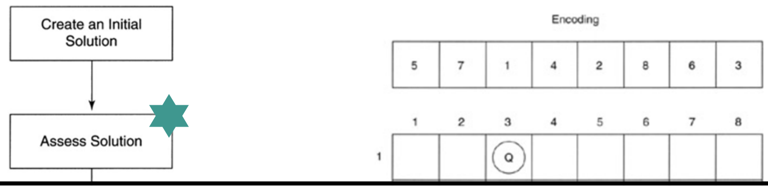


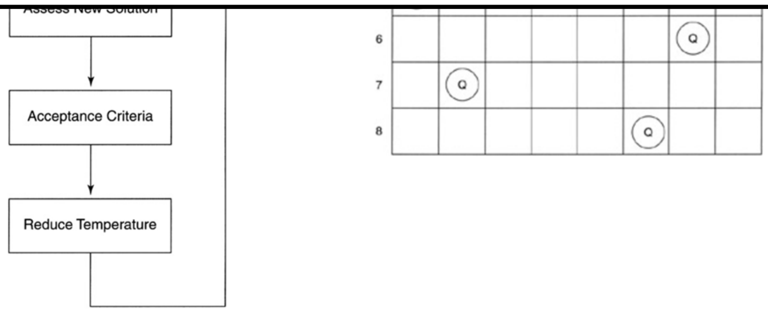
Figure 2.4: N-Queens solution encoding.

2장 모의 담금질, 예제: N-Queen 문제

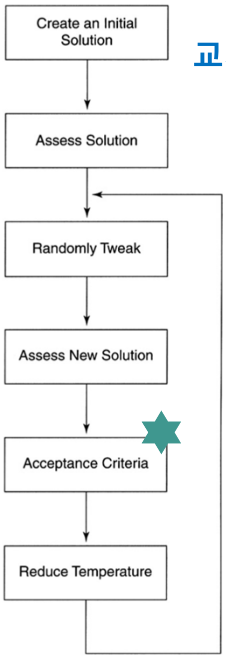


Energy

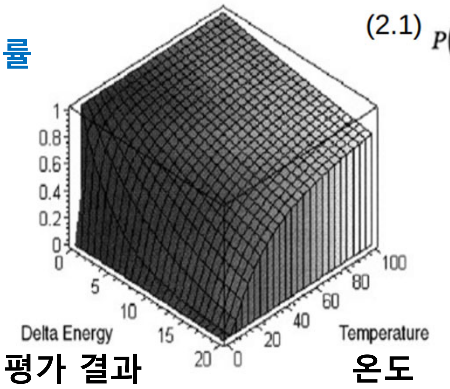
The energy of a solution is defined as the number of conflicts that arise given an encoding. The goal is to find an encoding that exhibits energy of zero, or no conflicts on the board.



2장 모의 담금질, 예제: N-Queen 문제



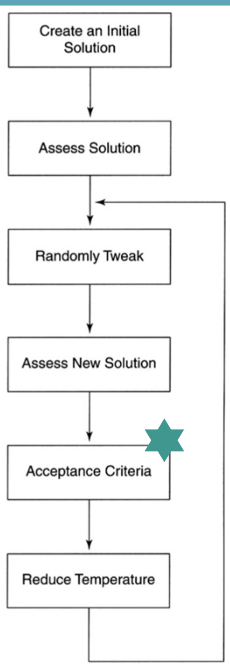
교체 확률

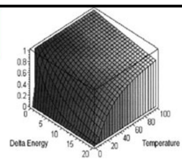


평가 결과 온도

$$P(\delta E) = \exp\left(-\frac{\delta E}{T}\right) \quad (2.1)$$

2장 모의 담금질, 예제: N-Queen 문제





$$P(\delta E) = \exp\left(-\frac{\delta E}{T}\right) \quad (2.1)$$

current solution energy 10

working solution energy 20

$$P = \exp\left(\frac{-10}{50}\right) = 0.818731$$

current solution energy 3

working solution energy 7

$$P = \exp\left(\frac{-4}{2}\right) = 0.135335$$

2장 모의 담금질, 코드

Listing 2.1: Types and Symbolic Constants.

```
#define MAX_LENGTH 30

typedef int solutionType[MAX_LENGTH];

typedef struct {
  solutionType solution;
  float energy;
} memberType;

/* Annealing Schedule */
#define INITIAL_TEMPERATURE      30.0
#define FINAL_TEMPERATURE        0.5
#define ALPHA                    0.99
#define STEPS_PER_CHANGE         100
```

2장 모의 담금질, 코드, 실행 결과

Sample Run

Let's look at the result of a sample run. In this run, we'll start the temperature at 100, though it's really only necessary to start at about 30 to solve the problem. We start from 100 to illustrate the algorithm (as shown in Figure 2.5).

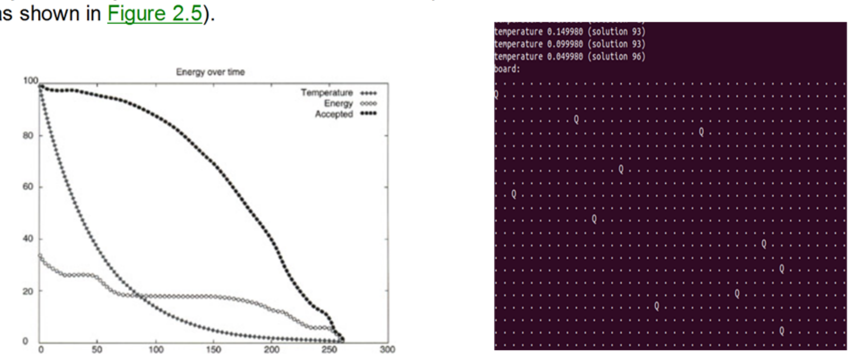


Figure 2.5: Plot of sample simulated annealing run for the 40-Queens problem.