Optimization algorithms inspired by nature

Mr.sc. Marko Čupić
Zagreb, June 2., 2011.
Introduction

- Emerging intelligence
- Optimization problems
- Genetic algorithm
- Ant colony optimization
Emerging intelligence

- Emerging intelligence
- Optimization problems
- Genetic algorithm
- Ant colony optimization
Introduction: Evolution
Introduction: Evolution

Evolution

(or is it?)
Introduction: Flock of birds
Introduction: Bea swarm
Introduction: Ant colonies

- Area of 50 m², depth of 8 m
Introduction: Ant colonies

- Ant mound, path...
Introduction: Ant colonies
Introduction: Ant colonies
Introduction: School of fish
Dumb parts, properly connected into a swarm, yield smart results.

Kevin Kelly
New Rules for the New Economy
Sep 1997
Introduction: conclusions

The **whole** is **greater** than the **sum** of the **parts**.
Optimization problems

- Emerging intelligence
- **Optimization problems**
- Genetic algorithm
- Ant colony optimization
Optimization problems

- Optimization: the procedure of finding the best solution of a problem, the solution with the lowest price
- Typically:
  - Continuous variables
  - Combinatorial problems
Optimization problems

• State space search:
  – Find a path from $S_0$ to $S_F$
  – The solution is the **path** (e.g. 3x3 jigsaw puzzle!)

• CSP: **Constraint Satisfaction Problem**
  – A kind of state space search where the path from $S_0$ to $S_F$ is not important. The solution is the final state itself.
Optimization problems

- CSP: Constraint Satisfaction Problem
  - Constraints that must be satisfied are defined
  - A criterion function to optimize is given
Optimization problems

• We’ve seen ways to tackle combinatorial problems
  – State space search algorithms
    • Breadth first search
    • Depth first search
    • A*
    • ...

• Unfortunately, often not applicable to real problems
Optimization problems

• The traveling salesman problem
  – Coordinates of n cities on the map are given
  – Find the shortest tour through all cities
  – Mathematically: find the shortest Hamiltonian cycle in the graph
  – **NP hard problem**
    (factorial complexity)
Optimization problems

- Traveling salesman problem
Optimization problems

- Traveling salesman problem
  - 12 cities, 12 sec
  - 13 cities, 2.5 min
  - 14 cities, 0.5h
  - 15 cities, 7.6h
  - 16 cities, 4.7 days
  - ...
  - 500 cities, ????
Optimization problems

• Other problems
  – Scheduling unscheduled students into groups for classes
  – Midterm timetable creation
  – Lab assignments timetable creation

• Enumerating all possibilities?
  – It would take much much more time than the age of the universe
Optimization problems

• Heuristics
  – Algorithms that find sufficiently good solutions, usually do not guarantee optimality, and have low computational complexity (polynomial)
  – They can be
    • Construction based
    • Local search based
Optimization problems

• Heuristics
  – Construction based
    • Build the solution incrementally
  – Local search algorithms
    • Start with a completed solution and try to incrementally make it better
Optimization problems

• Metaheuristics
  – A set of algorithmic concepts used to define heuristic methods applicable to a wide set of problems
  – A heuristic guiding problem specific heuristics
Optimization problems

• Metaheuristics
  – Simulated annealing
  – Taboo search
  – Evolutionary algorithms
  – Ant colony optimization
  – Swarm optimization
  – Artificial imunological systems
  – …
A problem 😊

• "No free lunch“ Theorem, Wolpert & Macready, 1995, 1997:

  – All algorithms seeking an optimum of a goal function behave identically with respect to any performance measure, when considered averaged over all possible goal functions

...
A problem 😊

• "No free lunch“ Theorem, Wolpert & Macready, 1995, 1997:

  – Specifically, if algorithm $A$ is better than algorithm $B$ on some goal functions, then, roughly speaking, there must be exactly that many different goal functions on which $B$ is better than $A$. 
Genetic algorithm

- Emerging intelligence
- Optimization problems
- **Genetic algorithm**
- Ant colony optimization
Genetic algorithm

• Evolution as inspiration
• Population based algorithms
• Darwins theory about the origin of species
Genetic algorithm

• Main settings: Darwin
  – Fertility of species – there are always more descendants than required
  – Size of the population is roughly constant
  – Food supply is limited
  – For species that reproduce sexually there are no identical individuals, there are variations
  – Most of an individuals specific variations is passed on to its descendants
Genetic algorithm

- Example problem \( f(x) = 10 + x^2 - 10 \cdot \cos(2 \pi x) \)
  - Find \( x \) for which \( f(x) \) is minimal
Genetic algorithm

• How does GA work?
  – There is a population of chromosomes
  – Each chromosome represents one solution to the problem
  – Each solution has a fitness
  – In our example fitness and \( f(x) \) are opposite ➔ higher \( f(x) \) means lower fitness
Genetic algorithm

• Implementation
  – From the current generation we iteratively create the next one
  – We select individuals that have a higher probability of creating better solutions
  – They are combined using the crossover operator
  – Resulting individuals are mutated using the mutation operator
Genetic algorithm

- Flowchart
Genetic algorithm

• Roles
  – **Selection** → selectional pressure → speed of convergence
  – **Crossover** → searching the neighbourhood of parents
  – **Mutation** → getting out of local optima, big jumps in the solution space
Genetic algorithm

• Binary chromosome 010001111
  – A sequence of binary digits interpreted as a solution (value of a variable)
  – Three bit chromosome: 000, 001, …, 111
  – Assuming we are observing a real variable from the interval \([-2, 2]\), then: 000\(\equiv\) -2, 001\(\equiv\) -1.43, …, 111\(\equiv\) 2
  – What is the number of bits for a given precision?
Genetic algorithm

- Binary chromosome
  - A more complex example
    - Solution for a function of three variables x, y, z
Genetic algorithm

- Crossover with one breaking point
  - Two parents are chosen
  - A breaking point is randomly chosen
  - Crossover is performed
Genetic algorithm

- Other types of crossover
  - Crossover with one breaking point
  - Crossover with $n$ breaking points
  - Uniform crossover
  - ...
Genetic algorithm

- Mutation operator
  - Mutation probability is given
  - Each bit is flipped with that probability

- Can introduce a huge change!
Genetic algorithm

• Selection of parents
  – Proportional selection - Roulette-wheel selection
  – More fitness of an individual means higher chances in the selection process

\[
probSel(i) = \frac{fit(i)}{\sum_{j=1}^{n} fit(j)}
\]
Genetic algorithm

- Selection of parents – proportional selection

\[ \text{len}(i) = \frac{\text{fit}(i)}{\sum_{j=1}^{n} \text{fit}(j)} \]
Genetic algorithm

\[ P = \text{create\_initial\_population}(\text{POP\_SIZE}) \]
\[ \text{evaluate}(P) \]
\[ \text{repeat\_until\_done:} \]
\[ \text{new\_population } P' = \emptyset \]
\[ \text{repeat\_while size}(P')<\text{POP\_SIZE} \]
\[ \text{select } R1 \text{ and } R2 \text{ from } P \]
\[ \{D1, D2\} = \text{crossover}(R1, R2) \]
\[ \text{mutate } D1, \text{ mutate } D2 \]
\[ \text{add } D1 \text{ and } D2 \text{ into } P' \]
\[ \text{end\_repeat} \]
\[ P = P' \]
\[ \text{evaluate}(P) \]
\[ \text{end\_repeat} \]
Genetic algorithm
Ant colony optimization

• Emerging intelligence
• Optimization problems
• Genetic algorithm

• Ant colony optimization
Ant colony optimization

• Ants exhibit interesting behavior
  – They successfully find the shortest path to food sources
Ant colony optimization

- Experiments
Ant colony optimization

• Explanation
  – While moving ants leave a feromon trail behind
  – An ant moves randomly, but it is more likely to go in the direction where the feromon trail is stronger
Ant colony optimization

- Directly applicable to problems described by graphs
- E.g. From 1, possible next are 2, 3, 4

\[ \tau_0 = \text{konst} \]

\[ p_{ij}^k = \begin{cases} 
\frac{\tau_{ij}^\alpha}{\sum_{l \in N_i^k} \tau_{il}^\alpha}, & \text{if } j \in N_i^k \\
0, & \text{if } j \notin N_i^k
\end{cases} \]
Ant colony optimization

• Ant System algorithm
  – Using heuristic information has additional benefits on performance

\[
\tau_0 = \frac{m}{C^{nn}}
\]

\[
p_{ij}^k = \begin{cases} 
\frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{l \in N_i^k} (\tau_{il}^\alpha \cdot \eta_{il}^\beta)}, & \text{ako } j \in N_i^k \\
0, & \text{ako } j \notin N_i^k
\end{cases}
\]
Ant colony optimization

• Ant System algorithm

```plaintext
repeat until not done
  repeat for each ant
    create solution
    evaluate solution
  end repeat
  evaporate_pheromons
  repeat for all_or_some ants
    update_pheromons
  end repeat
end repeat
```
Ant colony optimization

- **Procedure:** Create_solution
  - An ant starts from a node
  - With respect to the probabilities, a next node is chosen, then another, and so on until the ant reaches the last node

Uz α=1, β=2:

\[
p(4 \rightarrow 3) = 11.9\% \quad p(4 \rightarrow 6) = 23.7\% \quad p(4 \rightarrow 7) = 64.4\%
\]
Ant colony optimization

- Procedure: Evaluate_solution
  - Calculates the total path length
  - Moving from one node to another is usually associated with a cost (cities → distance)
Ant colony optimization

• **Procedure:** Evaporate_pheromons
  – Lowers pheromone trails on all edges by an amount

\[
\tau_{ij} \leftarrow \tau_{ij} \cdot (1 - \rho)
\]

  – Geometric progression!
  – Very costly (graph has many edges)
Ant colony optimization

- **Procedura:** Update_pheromons
  - Funkcija za odabranog mrava dodaje nove feromonske tragove iznosa:

  \[
  \Delta \tau_{ij}^k = \begin{cases} 
  1/C^k, & \text{if edge } i-j \text{ is on the path of ant } k \\
  0, & \text{otherwise}
  \end{cases}
  \]

  - Novo stanje je tada:

  \[
  \tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{m} \Delta \tau_{ij}^k
  \]
Ant colony optimization

Broj gradova: 30
Duljina: 1180.239

<table>
<thead>
<tr>
<th>Korak</th>
<th>Mrav</th>
</tr>
</thead>
<tbody>
<tr>
<td>Populacija</td>
<td>Automatski</td>
</tr>
</tbody>
</table>
Conclusion

• Algorithms inspired by nature – a very vivid area of research!
• Methods that can efficiently tackle problems that were previously unsolvable
• New algorithms emerging (e.g. Bee Colony Optimization, Intelligent Water Drops, …)
Links

• Video about an ant colony
Links

• Materials
http://java.zemris.fer.hr/nastava/ui/

• Implementations
http://java.zemris.fer.hr/nastava/ui/ev oAlg.zip