# Experiences With Teaching Adaptive Optimization to Engineering Graduate Students

**Alice E. Smith**
Department of Industrial Engineering
University of Pittsburgh
Pittsburgh, PA 15261 USA
aesmith@engrng.pitt.edu
www.pitt.edu/~aesmith

**Abstract - This paper discusses the first-hand experiences of the author in developing and teaching a graduate level engineering course in adaptive optimization methods inspired by nature. The paper discusses course content, textbooks and supplementary written material, software and computer projects, and grading and evaluation. This course has encouraged many students to pursue research in evolutionary computation, tabu search or simulated annealing, however it is continually being modified to reflect the many changes occurring in the field.**

## 1 Introduction

The advent of new computational techniques inspired by nature has resulted in the publication of many new books and journals, new organizations and conferences, and a new group of graduate students who want to learn about these techniques. Developing new courses at the graduate level is time consuming and challenging. For new courses in evolutionary computation and other adaptive optimization methods inspired by nature, development is complicated because the field is rapidly changing, few comprehensive textbooks exist, students with a wide variety of backgrounds and interests are attracted, and software implementation is an open issue. This paper will focus on a traditional university course with the main emphasis on graduate education. However, it must be noted that there are many non-traditional means of instruction in evolutionary computation. There are short courses presented by experts from both academia and industry, there are instructional video tapes, and there are tutorial sessions during many conferences.

A very few papers in the literature have addressed computational intelligence education or courses. Winston describes a course in neural networks designed for working engineers [20]. Lee and Edwards write about a course in power plant control that emphasizes techniques from computational intelligence including fuzzy systems, neural networks and expert systems at the Pennsylvania State University [14]. Fowler and Hudson of Kansas State University describe a distance learning course for electrical engineering students on fuzzy logic and neural networks [7]. Another course focusing on electrical engineering was described by Ribeiro and Rogers at Dordt College in Iowa [18]. This course covers the analysis of power signals (currents and voltages) using wavelets, genetic algorithms, expert systems, fuzzy logic and neural networks. The rest of the paper describes the course developed by the author in adaptive optimization.

## 2 Current Course

This course, entitled "Heuristic Optimization," is an advanced industrial engineering graduate elective in operations research and has been offered three times thus far, about two years apart each time. The title is meant to distinguish it from traditional exact approaches in optimization, mainly mathematical programming techniques. There are many heuristics in optimization, mostly problem-specific approaches, and this course might be better titled as "Adaptive Optimization" or "Meta-Heuristics."

The course covers many recent techniques in optimization that have been inspired by nature. The main topics are simulated annealing, genetic algorithms, evolutionary strategies and tabu search, in that order. Neural network optimization is not addressed as it is covered in a separate course ("Neural Networks and Industrial Applications") that has been developed and taught by the author. About one lecture each is spent on (a) miscellaneous related heuristics (e.g., GRASP, Scatter Search, The Great Deluge Algorithm), (b) hybrid approaches, (c) validation and comparison of results, and (d) handling constraints. Part of a lecture is dedicated to the theory of NP-Completeness and complexity analysis of algorithms. Appendix I includes the most recent syllabus of the course.

Both combinatorial and continuous optimization problems are addressed with emphasis placed on combinatorics. The book *Modern Heuristic Techniques for Combinatorial Problems* by Reeves [17] has been used, but is supplemented heavily by papers, especially for the evolutionary algorithms. Other possible texts are *Evolutionary Algorithms in Theory and Practice* by Baeck [1], *Genetic Algorithms in Search, Optimization and Machine Learning* by Goldberg [10], *Evolutionary Computation* by Fogel [5], *Evolution and*

*Optimum Seeking* by Schwefel [19], *Tabu Search* by Glover and Laguna [9], and *Genetic Algorithms and Engineering Design* by Gen and Cheng [8]. This last book is aimed at those in manufacturing systems, industrial engineering and operations research. None of these books are as comprehensive, however, as the Reeves book. A newer edited book of papers could be used as a comprehensive text. This is *Modern Heuristic Search Methods* by Rayward-Smith et al. [16]. Another recent book that could be used as a supplement is *Evolutionary Computation*: *The Fossil Record* [6] by Fogel for a historic perspective on the field.

It is this author's opinion that none of the available software for these techniques should be used for teaching as each optimization problem requires its own encoding and approach. The techniques are straightforward to code, and the course starts with the simplest one to code (simulated annealing). The course is homework and project oriented with no exams.

One homework assignment per each of the five modules (the general heuristic approach, simulated annealing, genetic algorithms, evolutionary strategies, tabu search) is made. (Appendix II includes the most recent set of homework assignments.) The homeworks involve the students coding and testing the approaches on several well-known problems. The quadratic assignment problem (QAP) is a good combinatorial study vehicle. It is easy to understand and has been thoroughly investigated in the literature using traditional techniques and the newer heuristics. The Nugent [15] test suite of problems ranges from 5 to 30 departments, providing nice scale-up testing. Another possible well known combinatorial study vehicle is the traveling salesman problem (TSP). For continuous optimization, a good-sized test problem is the six hump camelback function where $x$ lies between ±3 and $y$ lies between ±2. The objective is to minimize $z$. The global minimum lies at (-0.0898, 0.7126) where $z$ = -1.0316.

$$z = [4 - 2.1x^2 + x^4/3]x^2 + xy + [-4 + 4y^2]y^2$$

Besides the homeworks, each student is assigned a journal article to read. The student must prepare a very brief (about 10 minutes) oral description of the article – its objectives, methods, results and contributions – to present to the class. A one or two page summary giving the citation and the material in the oral presentation must be written and a copy is distributed to each class member. This journal assignment fulfills several educational objectives. First, the students are introduced to research issues in more depth as new applications and versions of the heuristics are covered. Second, the students are given practice in short oral presentations in a low-pressure environment. Third, the students learn a little about the critical of reading journal articles and the various journals that are relevant to this field. The articles are selected so that they pertain to current or very recent classroom material. If a student has a strong desire to select his or her own article, that is allowed.

This project aspect of this course has been well received, with a large percentage of students pursuing Masters or PhD theses in the area. Some projects have included design of reliable networks, optimizing cellular manufacturing layouts and tooling, curve fitting and system identification, using space filling curves for TSP, and variations of the minimum spanning tree problem. Some of these projects were publishable with a little additional work. Also, some formed the nucleus of substantial further work in the form of a thesis or dissertation. Some projects were dead ends, either because of an unpromising match between the problem and the solution approach, or because of student interest.

This course has been quite successful. The first term it was offered, only four students remained through the entire course. The most recent time it was taught, there were about 15 students, which is sizeable for an advanced elective. Student reviews of the course have been generally very positive. The necessity of programming has discouraged some students from taking the class, however there does not seem to be any reasonable or effective way around this barrier.

## 3 Anticipated Changes

One addition to the syllabus the next time this course is taught will be the subject of ant colony methods as founded by Dorigo and others [2-4]. This is an important new paradigm of evolutionary computation that involves cooperative behavior. A smaller section might be added on particle swarm techniques as put forth by Kennedy and others [11-13]. These have a similar theme of cooperation among agents, but balance individual and group behavior.

This author would like to encourage Colin Reeves to update and expand his landmark volume [17], or perhaps others would take on the challenge of combining the many adaptive heuristics inspired by nature into one, definitive volume that could be used in the classroom.

## Acknowledgment

The author gratefully acknowledges support of the National Science Foundation CAREER grant DMI 95-02134.

## Bibliography

[1] Baeck, Thomas, *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, New York, NY, 1996.

[2] Costa, D. and A. Hertz, "Ants can colour graphs," *Journal of the Operational Research Society*, **48**, 1997, 295-305.

[3] Dorigo, M., V. Maniezzo and A. Colorni, "The ant system: Optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, **1**, 1996, 29-41.

[4] Dorigo, M. and L. M. Gambardella, "Ant colony system: A cooperative learning approach ot the traveling salesman problem," *IEEE Transactions on Evolutionary*

*Computation*, 1, 1997, 53-66.

[5] Fogel, David B., *Evolutionary Computing*, IEEE Press, New York, NY, 1995.

[6] Fogel, David B., *Evolutionary Computation: The Fossil Record*, IEEE Press, New York, NY, 1998.

[7] Fowler, Eddie R. and William B. Hudson, "Distance education from the faculty perspective," *Proceedings of the 1994 24th Annual Conference on Frontiers in Education*, San Jose, CA, 1994, 583-585.

[8] Gen, Mitsuo and Runwei Cheng, *Genetic Algorithms and Engineering Design*, John Wiley, New York, NY, 1997.

[9] Glover, Fred and Laguna, Manuel, *Tabu Search*, Kluwer Academic Publishers, Norwell, MA, 1997.

[10] Goldberg, David E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA, 1989.

[11] Kennedy, James, "The particle swarm: Social adaptation of knowledge," *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation*, IEEE Press, 1997, 303-308.

[12] Kennedy, James, "Thinking is social: Experiments with the adaptive culture model," *Journal of Conflict Resolution*, **42**, 56-76.

[13] Kennedy, James and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, IEEE Press, 1997, 4104-4109.

[14] Lee, Kwang Y., "Curriculum for development of power plant intelligent distributed control," *Proceedings of the 1995 Annual ASEE Conference*, Anaheim, CA, 1995, 1065-1071.

[15] Nugent, C. E., T. E. Vollmann and J. Ruml, "An experimental comparison of techniques for the assignment of facilities to locations," *Operations Research* **16**, 1968, 150-173.

[16] Rayward-Smith, V. J. et al. editors, *Modern Heuristic Search Methods*, John Wiley & Sons, Chichester, 1996.

[17] Reeves, Colin R., editor, *Modern Heuristic Techniques for Combinatorial Problems*, John Wiley & Sons, New York, 1993.

[18] Ribeiro, Paulo F. and David A. Rogers, "Power electronics, power quality and modern analytical tools: the impact on electrical engineering education," *Proceedings of the 1994 24th Annual Conference on Frontiers in Education*, San Jose, CA, 1994, 448-451.

[19] Schwefel, Hans-Paul, *Evolution and Optimum Seeking*, John Wiley & Sons, New York 1995.

[20] Winston, Arthur W., "Use of neural networks in engineering applications and the education of engineering leaders," *Proceedings of the 1995 Annual ASEE Conference*, Anaheim, CA, 1995, 2828-2834.

## Appendix I Syllabus of Heuristic Optimization

Course Objectives: This course is a survey of the newer, most common heuristic search methods. The areas of focus will be simulated annealing, genetic algorithms, evolutionary strategies and tabu search. Other methods, such as neural networks and random methods, will be briefly covered. Both combinatorial and continuous optimization problems will be considered, with emphasis on combinatorics. The main techniques will be introduced, discussed critically and variations presented. Key papers from the literature, including applications, will be used. Students should gain knowledge of how and why these techniques work, when they should be applied and their relative merits to each other and to more traditional approaches, such as mathematical programming.

Supplemental Material: *Journal of Heuristics*, *IEEE Transactions on Evolutionary Computation*, *Computers & Operations Research*, *IIE Transactions*, *INFORMS Journal on Computing*, *Evolutionary Computation*, *Complex Systems* (Physics Library), *Annals of Operations Research*, *Proceedings of the International Conferences on Genetic Algorithms* (1987 - 1997), *Proceedings of the IEEE International Conferences on Evolutionary Computation* (1994 - 1997), *Genetic Algorithms in Search, Optimization and Machine Learning* (Goldberg), *Adaptation in Natural and Artificial Systems* (Holland), *Genetic Programming* (Koza), *Handbook of Genetic Algorithms* (Davis), *Genetic Algorithms + Data Structures = Evolution Programs* (Michalewicz), *Simulated Annealing and Boltzmann Machines* (Aarts and Korst), *Genetic Algorithms and Simulated Annealing* (Davis), *Modern Heuristic Search Methods* (Rayward-Smith, et al., editors), *Evolutionary Computation* (Fogel). Most of these books are in the Engineering Library.

Required Skills: Programming of some sort (C, Basic, Pascal, etc.) is required to implement the optimization methods. This can be done on PC's, workstations or mainframes (VAX or UNIX) without extensive or sophisticated programming knowledge. Emphasis is on effectiveness, not computational efficiency in terms of CPU effort.

Course Structure: This course is an advanced graduate course with emphasis on self exploration and research. There will be homework assignments, a paper review and a project. The project can be done individually or in groups of two. The project can synthesize multiple techniques or be an in depth exploration of one technique using problems and applications of the student's choice. These techniques are flexible enough to accommodate a variety of applications in classic OR, manufacturing systems, engineering design and finance.

Grading: Each homework will be worth 10%, paper reviews will be worth 10% and the project will be worth 40%.

Journal Article Review: each student will randomly be given an article from one of the subjects we cover in class. Each student will be responsible for providing an oral tutorial on the article to the class during the appointed class period. Tell why this article is relevant to what is currently being discussed in class. Explain where and when it was published, and anything you know about the author(s). What was the aim of the article? Who was the intended audience? What research was done? How was it presented? Did the results support the premise of the article? Was the argument convincing, or did more or better work need to be done? How could the results of this article be used? How much of a contribution to the field does it make? Any other reactions you have to the article. If you have a strong preference to choose your own article, please discuss it with me and this can be arranged.

**Schedule of Classes**

| Date | Subject | Assignments Due |
|---|---|---|
| 1/15 | Introduction to Heuristic Search | |
| 1/22 | Simulated Annealing | HWK 1 (Intro) |
| 1/29 | Simulated Annealing | |
| 2/5 | Evolutionary Computation | HWK 2 (SA) |
| 2/12 | Genetic Algorithms | |
| 2/19 | Genetic Algorithms | |
| 2/26 | Evolutionary Strategies | HWK 3 (GA) |
| 3/12 | Tabu Search | |
| 3/19 | Tabu Search | HWK 4 (ES) |
| 3/26 | Tabu Search / NP Completeness | |
| 4/2 | Constraint Handling | HWK 5 (TS) |
| 4/9 | Other and Hybrid Methods | |
| 4/16 | Validation and Comparison of Results | |
| 4/23 | Project Presentations | Project |

## Appendix II Heuristic Optimization Homework Assignments

## Homework Assignment 1

Problem One

There is one freight elevator at Benedum Hall that has a capacity of 640 cubic feet. You have a shipment of items in boxes that need to be transferred from the ground floor upwards. Some boxes go to the Auto ID Lab on the fifth floor and some go to the IE Department on the 10th floor. There are five different sizes of boxes ranging from 2.5 cubic feet to 30 cubic feet and there is a total volume of boxes of 2000 cubic feet.

a. Formulate three possible objective functions to optimize given this situation and state any assumptions you have made. For each objective, note the constraints involved, and whether they are hard or soft constraints.

b. Select one objective function from part a, and formulate and describe a deterministic heuristic procedure to solve this optimization problem.

c. Select one objective function from part a, and formulate and describe a stochastic heuristic procedure to solve this optimization problem.

Problem Two

A discrete event SIMAN simulation model of an (S,s) System has two input (decision) variables - inventory level when reorder should take place (s) and the number of items to order (S-s) - and one output of interest, total cost of the inventory system (c). The objective is to set values of the two decision variables, which can take on any values within certain ranges of each variable, to minimize the total cost of the inventory system. You can regard the simulation as producing a continuous, though not necessarily smooth, response surface over all values of the decision variables within the predefined ranges. An important concern is the computer time needed to make the simulation runs, so the optimization strategy adopted should conserve the number of simulation runs made (i.e., number of possible solutions searched).

a. Formulate and describe a deterministic heuristic procedure to solve this minimization problem.

b. Formulate and describe a stochastic heuristic procedure to solve this minimization problem.

Problem Three

In heuristic search, one of the primary goals is to properly balance global search and local search. Discuss how this balance is problem dependent, in terms of size of the search space, shape of the response surface, constraints imposed on the problem, difficulty or ease of calculating the objective function, and user requirements for the near-optimality of the final solution. Devise a global search operator or move and a local search operator or move for each of the two problems above (the elevator packing and the simulation).

Problem 4

One of the motivations for using a heuristic method for optimization is the size of the search space of the problem. Discuss ways in which the size of the search space for (a) combinatorial and (b) continuous problems can be calculated or estimated.

## Homework 2

For the below two problems do the following:

1. Code a simple SA to solve the problem. To do this you need to encode the problem, develop the definition of a neighborhood, define a move operator, select an annealing schedule and select a stopping criterion.
2. Run your SA.
3. Perform the following changes on your SA code (one by one) and compare the results.
   - change the annealing schedule twice
   - change the stopping criterion twice
   - change the initial starting point (initial solution) five times
   - change the random number seed ten times
4. Please include your code with your homework.
5. I am including the global optimum to each problem, but don't use it in your solution methodology.

Problem 1

This is the smallest of the QAP (quadratic assignment problem) test problems of Nugent et al. Five departments are to be placed in five locations with two in the top row and three in the bottom row (see below). The objective is to minimize costs between the placed departments. The cost is (flow * rectilinear distance), where both flow and distance are symmetric between any given pair of departments. Below is the flow and distance matrix where distance is the upper half. The optimal solution is 25 (or 50 if you double the flows).

| Dept | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | - | 1 | 1 | 2 | 3 |
| 2 | 5 | - | 2 | 1 | 2 |
| 3 | 2 | 3 | - | 1 | 2 |
| 4 | 4 | 0 | 0 | - | 1 |
| 5 | 1 | 2 | 0 | 5 | - |

Problem 2

This is the six hump camelback function where *x* lies between ±3 and *y* lies between ±2. The objective is to minimize *z*. The global minimum lies at (-0.0898, 0.7126) where *z* = -1.0316.

Problem 3

Discuss how you might decide whether a problem is appropriate for simulated annealing. If it is appropriate, how would you decide on a move operator and an annealing schedule? If you implemented these and the results were poor, what alterations would you try next?

## Homework 3

For the below problems do the following:

1. Code a simple GA to solve the problem. To do this you need to encode the problem as a binary string (first problem) or permutation (second problem), use single point or uniform crossover and bit flip or 2-opt mutation, use a form of roulette wheel selection or tournament selection, set a population size and select a stopping criterion.
2. Run your GA.
3. Perform the following changes on your GA code (one by one) and compare the results.
   - change the initial starting points (initial solutions) 10 times
   - change the mutation probability twice
   - change the population size twice
4. Please include your code with your homework.

Continuous Optimization Problem

A: This problem is a well known hub location problem where the objective is to minimize the sum of the Euclidean distances between the hub site (in 2 coordinate space) and the coordinates of 25 large U.S. cities. You can ignore curvature of the earth. The decision variables are the *x* and *y* coordinates of the hub site. The 25 city coordinate matrix is listed on the attached sheet.
B: Redo this problem but use two hub sites. This requires not only minimizing the two summations of distance, but also assigning each of the 25 cities to one or the other hub.

Problem 2

Use a permutation based GA to solve the QAP problem. Since the search space is small, this GA should use a relatively small population size. This is the fourth of the QAP (quadratic assignment problem) test problems of Nugent et al. Eight departments are to be placed in eight locations with four in the top row and four in the bottom row. The objective is to minimize flow costs between the placed departments. The flow cost is (flow * distance), where both flow and distance are symmetric between any given pair of departments. Below is the flow and distance matrix where rectilinear distance is the upper half. The optimal solution is 107 (or 214 if you double the flows).

| Dept | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | - | 1 | 2 | 3 | 1 | 2 | 3 | 4 |
| 2 | 5 | - | 1 | 2 | 2 | 1 | 2 | 3 |
| 3 | 2 | 3 | - | 1 | 3 | 2 | 1 | 2 |
| 4 | 4 | 0 | 0 | - | 4 | 3 | 2 | 1 |
| 5 | 1 | 2 | 0 | 5 | - | 1 | 2 | 3 |
| 6 | 0 | 2 | 0 | 2 | 10 | - | 1 | 2 |
| 7 | 0 | 2 | 0 | 2 | 0 | 5 | - | 1 |
| 8 | 6 | 0 | 5 | 10 | 0 | 1 | 10 | - |

## Homework 4

For the six hump camelback function below do the following:

1. Code a simple ES to solve the problem. To do this you need to encode the problem as two real variables, select values for $\mu$ and $\lambda$, select a standard deviation (the same for each variable) and change it using the 1/5 rule, randomly select an initial population and select a stopping criterion. Do not perform recombination.
2. Run your ES for ten different seeds.
3. Perform the following changes on your ES code (do this cumulatively) and compare the results.
   - change the ratio of $\mu$ to $\lambda$
   - add individual standard deviations for each variable to the encoding and alter them using global intermediate recombination
   - add recombination to the two variables using dual discrete recombination
4. Please include your code with your homework.

## Homework 5

For the 15 department QAP from Nugent (below) do the following:

1. Code a simple TS to solve the problem. To do this you need to encode the problem as a permutation, define a neighborhood and a move operator, set a tabu list size and select a stopping criterion. Use only a recency based tabu list and no aspiration criteria at this point.
2. Run your TS.
3. Perform the following changes on your TS code (one by one, being cumulative) and compare the results.
   - change the initial starting solution 5 times
   - change the tabu list size - smaller and larger than your original choice
   - change the tabu list size to a dynamic one - an easy way to do this is to choose a range and generate a random uniform integer between this range every so often (i.e., only change the tabu list size infrequently)
   - add one or more aspiration criteria such as best solution so far, or best solution in the neighborhood, or in a number of iterations
   - use less than the whole neighborhood to select the next solution
   - add a frequency based tabu list and/or aspiration criteria (designed to encourage the search to diversify)
4. Please include your code with your homework.

This is the sixth of the QAP (quadratic assignment problem) test problems of Nugent et al. Fifteen departments are to be placed in 15 locations with five departments (columns) in three rows. The objective is to minimize flow costs between the placed departments. The flow cost is (flow * distance), where both flow and distance are symmetric between any given pair of departments. Below is the flow and distance matrix where rectilinear distance is the upper half. The optimal solution is 575 (or 1150 if you double the flows).

| Dept | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | - | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 5 | 2 | 3 | 4 | 5 | 6 |
| 2 | 10 | - | 1 | 2 | 3 | 2 | 1 | 2 | 3 | 4 | 3 | 2 | 3 | 4 | 5 |
| 3 | 0 | 1 | - | 1 | 2 | 3 | 2 | 1 | 2 | 3 | 4 | 3 | 2 | 3 | 4 |
| 4 | 5 | 3 | 10 | - | 1 | 4 | 3 | 2 | 1 | 2 | 5 | 4 | 3 | 2 | 3 |
| 5 | 1 | 2 | 2 | 1 | - | 5 | 4 | 3 | 2 | 1 | 6 | 5 | 4 | 3 | 2 |
| 6 | 0 | 2 | 0 | 1 | 3 | - | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 5 |
| 7 | 1 | 2 | 2 | 5 | 5 | 2 | - | 1 | 2 | 3 | 2 | 1 | 2 | 3 | 4 |
| 8 | 2 | 3 | 5 | 0 | 5 | 2 | 6 | - | 1 | 2 | 3 | 2 | 1 | 2 | 3 |
| 9 | 2 | 2 | 4 | 0 | 5 | 1 | 0 | 5 | - | 1 | 4 | 3 | 2 | 1 | 2 |
| 10 | 2 | 0 | 5 | 2 | 1 | 5 | 1 | 2 | 0 | - | 5 | 4 | 3 | 2 | 1 |
| 11 | 2 | 2 | 2 | 1 | 0 | 0 | 5 | 10 | 10 | 0 | - | 1 | 2 | 3 | 4 |
| 12 | 0 | 0 | 2 | 0 | 3 | 0 | 5 | 0 | 5 | 4 | 5 | - | 1 | 2 | 3 |
| 13 | 4 | 10 | 5 | 2 | 0 | 2 | 5 | 5 | 10 | 0 | 0 | 3 | - | 1 | 2 |
| 14 | 0 | 5 | 5 | 5 | 5 | 5 | 1 | 0 | 0 | 0 | 5 | 3 | 10 | - | 1 |
| 15 | 0 | 0 | 5 | 0 | 5 | 10 | 0 | 0 | 2 | 5 | 0 | 0 | 2 | 4 | - |