ABSTRACT
We propose a new modular genetic programming for finding attractive and statistically sound technical rules. We restrict the problem space using well-known technical rules to discover attractive technical rules. Experimental results show our modular genetic programming can successfully find unknown attractive technical rules for Korean stock market.

Categories and Subject Descriptors
J.m [Computer Applications]: Miscellaneous

General Terms
Experimentation

Keywords
stocks, patterns, technical rules, modular genetic programming

1. INTRODUCTION
Technical pattern analyses [1] aim to capture appropriate timing in a stock market. However, most of them are based on investors’ limited intuitions.

Most of the previous studies are limited as well since they have focused on high returns of technical rules only. In other words, they try to find trading rules when to buy and sell irrespective of the complexity of the rules. If technical rules are complex, they generally match few cases, which degrades the generality even if they are highly profitable. In this context, we try to find attractive technical rules considering simplicity and generality as well as profitability. We focus only on buy side. Investigation for the sell side is symmetric to that for the buy side; thus, the method in this paper would be easily applicable.

In this paper, we propose a new modular genetic programming for finding attractive technical rules. Combining modules in a predetermined set, which are preserved during genetic evolution, we could reduce the problem space effectively. We found a number of new attractive technical rules in Korean stock market using our new modular genetic programming.
We also define a technical rule \( r \) to be frequent if \( |R(r)| \geq m \), where \( |R(r)| \) is the cardinality of the set \( R(r) \). The \( m \) and \( M \) are predetermined constants that require tuning by prior knowledge.

The fitness of a technical rule \( r \) is defined by:

\[
f(r) = \begin{cases} 
\frac{1}{n} \sum_{k=1}^{n} E_k(r) & \text{if } |r| < M, |R(r)| \geq m \\
0 & \text{otherwise}
\end{cases}
\]

where \( n \) is the number of consecutive days for smoothing of the expected earning rates. We call \( n \) the smoothing constant for \( f(r) \). Our problem is a maximization problem of finding a technical rule \( r \) that maximizes \( f(r) \).

3. GENETIC PROGRAMMING FRAMEWORKS

Since typical or unrestricted GPs are known to be inefficient to find valid solutions, we define a set of module rules and tried to find more attractive technical rules by recombination of them. The set of module rules includes several hundreds of well known technical rules such as white or black bodies, gap ups or downs, white or black marubozu, arrangements, up or down trends, stochastics and so on. For example, a white marubozu is represented by \( p_{w} (t) = p_{o} (t) \land p_{b} (t) = p_{l} (t) \land m \land p_{h} (t) > p_{l} (t) \), where \( m \) is a constant for describing the length of its body.

We use a steady-state genetic programming and set the smoothing constant \( n \) for \( f(r) \) and the minimum cardinality of \( R(r) \), or \( m \) to be 5 and 1000, respectively.

**Selection, Crossover, and Mutation:** The tournament selection is used. The crossover chooses two subtrees from parents at random and it swaps them. It is important that the crossover does not disrupt modules and preserves them as encapsulated units. We used a mutation that picks a module at random and replaces it with another module chosen at random.

**Replacement and Stopping Criterion:** We replace the worst individual and stop if the number of generations reaches 1000.

**Local Optimization:** We traverse each module in an individual and compute the gain if it is replaced with another module chosen at random. We apply the replacement of module only if the maximum gain is positive.

4. EXPERIMENTAL RESULTS

We tested our GP with Korean stocks without funds and preferred stocks. We trained our GP with three consecutive years and tested the solutions with the next year. This process was shifted year by year.

For easier interpretations, we define \( f'(r) \) to be percentage of returns as follows:

\[
f'_s(r) = \left( f_s(r) - 1 \right) \times 100
\]

where \( f_s(r) \) is the fitness of rule \( r \) on set \( S \). \( S \) is either the training set \( T \) or the test set \( D \). The \( f' \) is called fitness in terms of return percentage. The best rule on training set will be denoted by \( r_b \) and the sets of matched cases by \( r_b \) on training set \( T \) and test set \( D \) are denoted by \( R_T (r_b) \) and \( R_D (r_b) \), respectively.

Table 2 shows our experimental results. The profit rates and the set sizes of the matched cases were averaged over 50 independent runs.

Our GP found highly attractive rules with average returns higher than 10% on most test sets except for the year 2004. The average return of rules found in all the test year was 15.6%. The return of rules in each test year over that of the corresponding training years, i.e., \( f'_{D}(r_b)/f'_{T}(r_b) \), is 0.98 on average, which means the GP is free from the problem of overtraining.

Table 3 shows the best rules for some test years. One can see that the GP found out significantly different rules from those known to be effective. We found that our best rules were mostly the combinations of gap up rules and moving averages. It is notable that the best rules were not complex.

This implies that attractive technical rules are generally not very complex.

5. CONCLUSIONS

We proposed a modular genetic programming for finding attractive technical rules. We defined the properties for attractive technical rules and tried to find them using a modular GP. Experimental results showed that our GP successfully found attractive rules for Korean stock market.

6. ACKNOWLEDGMENTS

This work was supported by the Brain Korea 21 Project. The ICT at Seoul National University provided research facilities for this study.

7. REFERENCES