2016 International Conference on Business and Information
July 3-5, 2016, Nagoya, Japan

EVIDENCE AGAINST COMBINING TECHNICAL TRADING RULES USING PARTICLE SWARM OPTIMIZATION

Sumedh Chalermkanjana,
Faculty of Commerce and Accountancy, Chulalongkorn University,
254 Pathumwan, Bangkok, Thailand, 10330.
pty1196@yahoo.com

Kotchakorn Chalermkanjana,
Faculty of Commerce and Accountancy, Chulalongkorn University,
254 Pathumwan, Bangkok, Thailand, 10330.
kotchakorn@cbs.chula.ac.th

ABSTRACT

Wang, Yu, Cheung, 2014a proposed complex trading strategy called Performance-based Reward Strategy (PRS). PRS combines component rules of 121 moving average (MA) rules and 19 trading range breakout (TRB) rules. To find the optimal set of parameters for PRS, the researchers used time variant particle swarm optimization (TVPSO) algorithm. The objective of the experiment was to measure annual net profit (ANP) of PRS for trading constituents of NASDAQ100. Wang, Yu, Cheung, 2014b conducted the same experiment but used seven classes of technical trading rules resulting in 1,059 component trading rules. For both of these studies, the researchers conducted the experiment only one time and PRS's ANP was better than all component rules'. Thus both of these studies concluded that PRS outperforms all component rules. TVPSO algorithm is an approximation algorithm and running TVPSO only once may bias the conclusion. To improve reliability of the conclusion, we conduct both experiments 100 times. For both experiments, we find that mean ANP of PRS was lower than that of the best of the best component rules. Furthermore, when we replicate PRS, we find that weight updating equation proposed in both of these studies was wrong and we propose weight updating equation which we believe both of these studies used. Furthermore, to speed up the running time of TVPSO, a simple parallel programing idea for it is proposed.

Keyword: Machine Learning, Technical Trading Rules, Swarm

0. PREREQUISITE

To avoid repeating ideas presented in both of these studies, readers are required to read Wang et al, 2014a and b.

1. INTRODUCTION

Wang et al, 2014a proposed a complex stock trading strategy called Performance-based Reward Strategy (PRS). PRS combines the two most popular classes of technical trading rules – moving average (MA) and trading range break-out (TRB). For both MA and TRB, PRS includes various combinations of the rule parameters to produce a universe of 140 component trading rules in all. Each component rule is assigned a starting weight, and a reward/penalty mechanism based on rules' recent profit is proposed to update their weights over time. To determine the best parameter values of PRS, it employs improved time variant particle swarm optimization (TVPSO) algorithm with the objective of maximizing the annual net profit (ANP) generated by PRS.

In Wang et al, 2014b, the same authors expanded the scope to combine the seven most popular classes of trading rules in financial markets, resulting in a total of 1,059 component rules. Beside MA and TRB, other component rules are Bollinger Bands (BB), Relative Strength Index (RSI), Stochastic Oscillator (STO), Moving Average Convergence/Divergence (MACD) and On-Balance Volume Average (OBVA). Due to a large number of component rules and swarm size, the optimization time was significant. A parallel PSO (particle swarm optimization) based on Hadoop was employed to optimize PRS more efficiently.

Ratnaweera et al, 2004 ran PSO-TVAC to solve f_1 and f_3 functions for 50 trials each. See Ratnaweera et al, 2004 for details of f_1 and f_3 functions. 50 trials were needed because PSO-TVAC is an approximation algorithm which means it may not find optimal solutions in a particular run. Therefore, in this study, we decide to rerun experiments in Wang et al, 2014a and b for 100 times each.

The rest of this study is organized as follows. Section 2 presents experiment critique and note. Section 3 presents parallel programming idea. Section 4 presents weight updating equation. Section 5 presents component rule replication. Section 6 presents results of experiment 1. Section 7 presents results of experiment 2. Section 8 presents conclusion.

2. EXPERIMENT CRITIQUE AND NOTE

Parameter tuning is used in machine learning to tune a particular algorithm to beat benchmark. One downside of parameter tuning is if input data is changed, the same set of parameters may not result in algorithm beating the benchmark. The configurations of Wang et al, 2014a and b are a little different. Wang et al, 2014a used boundaries for buy threshold as [0, 0.9] and for sell threshold as [-0.9, 0]. Wang et al, 2014b used boundaries for buy threshold as [0, 0.3] and for sell threshold as [-0.3, 0]. There was no reason given in Wang et al, 2014b why boundaries were changed. We suspect that in Wang et al, 2014b the buy threshold and sell threshold boundary change is parameter tuning.

Wang et al, 2014a and b used boundary for alpha as [-1, 1]. They gave reason for this boundary range as to avoid overfitting of training PRS. We suspect this boundary for alpha is parameter tuning.

In this study, we replicate Wang et al, 2014a as conducting experiment 1 and Wang et al, 2014b as conducting experiment 2. All the results shown for experiment 1 and 2 are for data in the testing period.

3. PARALLEL PROGRAMMING IDEA

In experiment 1, the highest computational complexity component is PRS. At the worst case scenario, where all the particles are evaluated by PSO in experiment 1, there are 100,000 particles. Thus, we decide to parallelize the Swarm evaluation of fitness function which is PRS. In this study, we use Java as the programming language. We use Java Thread to ease parallel programming implementation. The idea used for parallel programming is as in Table 1.

Table 1: Algorithm for paralleling PSO

Assume x is the number of cores. Assume number of particle in PSO is num_p. Let y = num_p / x Create y number of threads. Divide num_p particles into y equally sized group. Let each thread evaluate fitness function for its particle group.

4. PROFIT DETERMINATION (WEIGHT UPDATING EQUATION)

Wang et al, 2014a and b states that PRS increases weight of component rule with $P_i > 0$ (P_i = profit of component rule i) and penalize weight of component rule with $P_i < 0$. Wang et al, 2014a did not explain how they calculated profit exactly. We contacted the authors asking for profit calculation procedure and received no response from them.

After reading Wang et al, 2014a, we understand that P_i is calculated as in equation 2.

Equation 2: Profit determination interpreted from reading Wang et al, 2014a

P_i = {equity value at the end of day (t – 1)} – {equity value at the beginning of day (t – memory span)}

We input the optimal parameters from Wang et al, 2014a into PRS and run PRS to obtain the statistics. We also run another PRS with these optimal parameters but we set reward factor to be 0.0. This makes PRS to become weighted strategy (WS). The performance of PRS and WS with parameters set to the optimal values of PRS from Wang et al, 2014a are in Table 3. In Table 3, PRS has lower ANP than WS. This means that the reward/penalty mechanism of PRS does not add value above WS strategy. No. Trades of PRS is 339 which is 30 trades higher than reported in Wang et al, 2014a.

Table 3: Performance of PRS and WS with parameters set to optimal values of PRS from Wang et al, 2014a in the testing period

Trading rule	ANP (%)	Sharpe ratio	Payoff ratio	No. trades	Win%
PRS	19.5812	0.9983	5.0318	339	47.79
WS	21.3541	1.0860	6.6496	331	48.94

The profit determination of PRS should be intelligent so that PRS will adapt to various time periods. Consider the time period tp1: from day (t – memory span) to day (t – 1). We propose P_i to be calculated as in equation 4.

Equation 4: Our proposed profit calculation for reward/penalty mechanism of PRS

$$P_i = P\_CR_i - P\_PRS_i$$

Where

P_CR_i = {equity value at the end of day (t – 1) for component rule i}
– {equity value at the beginning of day (t – memory span) for component rule i}

P_PRS_i = {equity value at the end of day (t – 1) for PRS}
– {equity value at the beginning of day (t – memory span) for PRS}

The intuition for equation 4 is PRS will always increase the weights of those component rules which have ANPs greater than its ANP and penalize other rules that have lesser ANPs than ANP of PRS. In other words, PRS always put more weight for winner rules. The performance of PRS with our proposed profit determination is in Table 5. The ANP of PRS is 21.7666% which is higher than ANP of WS from Table 3. This means that reward/penalty mechanism of PRS adds value above WS strategy. The ANP of 21.7666% is similar to ANP of PRS of 22.2210% which is reported in Wang et al, 2014a. Number of trade is 290 which is lower than that reported in Wang et al, 2014a by 20 trades. This difference of 20 trades is lower than difference of 30 trades of PRS which is reported in Table 3. Therefore, we believe that profit calculation from Wang el, 2014a may be as in our proposed profit determination.

Table 5: Performance of PRS with our proposed profit determination and parameters set to optimal values of PRS from Wang et al, 2014a in the testing period.

Trading rule	ANP (%)	Sharpe ratio	Payoff ratio	No. trades	Win%
PRS	21.7666	1.0224	7.2398	290	48.97

5. COMPONENT RULE REPLICATION

To perform experiment 1 and 2, we need to replicate component rules. Table 6 shows performance of the seven best component rules in the testing period for transaction cost C = 0.001 produced in our study and in Wang et al, 2014b. In table 6 and for the rest of the study, the best component rules are represented as MA (nl, ns), TRB (n), BB (n, k), RSI (n, ob, os), STO (n, m, ob, os), MACD (nl, ns, m) and OBVA (nl, ns). For MA, nl and ns are parameters of MA. See Wang et al, 2014b for explanation of parameters of each component rule. We have trouble replicating the following component rules: BB, RSI, MACD. Our criteria for replicating component rules is to minimize difference in ANP when compared with Wang et al, 2014b. The trouble can be seen from the large difference of ANP of each of these component rules when compared with values reported in Wang et al, 2014b (e.g. Difference of the Best BB is -7.2851). The reason for not being able to replicate closely component rules may come from the difference in actual implementation that the authors used in Wang et al, 2014b of these rules than stated in their studies. Because there is no trouble replicating MA and TRB, we can replicate closely component rules for experiment 1. The trouble in replicating closely other classes of component rules effects results of experiment 2.

Table 6: Performance of the seven best component rules in the testing period for transaction cost C = 0.001 produced in our study and in Wang et al, 2014b.

	ANP (%)	Sharpe ratio	Payoff ratio	No. of trades	Win%
Best MA (150-125)	18.7529	1.0948	3.5056	524	54.9618
Best MA* (150-125)	18.8000	1.0700	3.5300	531	54.4000
Difference of Best MA	-0.0471	0.0248	-0.0244	-7	0.5618
Best TRB (125)	16.1374	1.1138	5.4719	274	54.3796
Best TRB* (125)	16.0000	1.0500	5.8400	276	52.2000
Difference of Best TRB	0.1374	0.0638	-0.3681	-2	2.1796
Best BB (10-1.9)	11.3149	1.0474	0.6984	2769	70.6031
Best BB* (30-2.3)	18.6000	1.1400	4.6200	627	49.8000
Difference of Best BB	-7.2851	-0.0926	-3.9216	2142	20.8031
Best RSI (19-80-20)	20.9548	0.9398	5.7671	343	48.1050
Best RSI* (13, 80, 30)	9.0000	0.6000	0.8100	801	70.9000
Difference of Best RSI	11.9548	0.3398	4.9571	-458	-22.7950
Best STO (10-7-90-20)	11.2678	0.7709	1.6144	259	72.9730
Best STO* (10-3-90-20)	11.6000	0.7600	0.7400	1401	71.7000
Difference of Best STO	-0.3322	0.0109	0.8744	-1142	1.2730
Best MACD (100-20-9)	15.9618	0.9361	3.7563	1350	35.2593
Best MACD* (100-40-15)	11.4000	0.8200	2.7500	1674	38.5000
Difference of Best MACD	4.5618	0.1161	1.0063	-324	-3.2407
Best OBVA (75-50)	10.4040	0.8095	1.3686	2215	54.9887
Best OBVA* (75-50)	10.7000	0.7200	1.4900	2157	53.7000
Difference of Best OBVA	-0.2960	0.0895	-0.1214	58	1.2887

* These are values obtained from Wang et al, 2014b.

6. RESULTS OF EXPERIMENT 1

Table 7, 8, 9 show performance of experiment 1 for PRS, WS and the two best component rules in the testing period for transaction cost C = 0.001, 0.002, 0.005 respectively. Wang et al, 2014a and b used transaction cost C = 0.001 as the main case for comparing performance of PRS with other benchmarks. In table 7, PRS's average ANP is less than the best MA's ANP. Furthermore, PRS's average ANP is greater than the best TRB's ANP and WS's average ANP. PRS's average ANP declines as C increases from 0.001 to 0.005.

Table 7: Performance of experiment 1 for PRS, WS and the two best component rules in the testing period for transaction cost C = 0.001. PRS and WS are run for 100 times.

	PRS	WS	Best MA (150-125)	Best TRB (125)
ANP (%)	16.9684	15.3411	18.7529	16.1374
	(1.8119)	(0.7462)		
Sharpe ratio	0.9686	0.9350	1.0948	1.1138
	(0.0411)	(0.0189)		
Payoff ratio	6.0440	5.8425	3.5056	5.4719
	(0.6660)	(0.4438)		
No. of trades	205.6900	205.5500	524	274
	(25.8569)	(13.9880)		
Win%	54.4525	53.0424	54.9618	54.3796
	(2.1821)	(1.1982)		
Profitable stock count	35.2700	32.5300	34	41
	(2.3862)	(1.5139)		
Non-profitable stock count	16.7300	19.4700	18	11
	(2.3862)	(1.5139)		
Profit ratio	67.8269	62.5577	65.3846	78.8462
	(4.5889)	(2.9114)		
Profitable stock net profit	13.5131	11.5292	15.8071	12.2233
(millions)	(2.3643)	(0.8428)		
Non-profitable stock net profit	-0.3680	-0.4227	-0.4416	-0.2139
(millions)	(0.0767)	(0.0387)		
Stock net profit	13.1451	11.1065	15.3655	12.0094
(millions)	(2.4191)	(0.8350)		

Table 8: Performance of experiment 1 for PRS, WS and the two best component rules in the testing period for transaction cost C = 0.002. PRS and WS are run for 100 times.

	PRS	WS	Best MA (150-125)	Best TRB (125)
ANP (%)	16.8767	15.3007	18.5163	16.0151
	(1.7900)	(0.7416)		
Sharpe ratio	0.9593	0.9360	1.0829	1.1064
	(0.0431)	(0.0202)		
Payoff ratio	5.9793	5.8654	3.6293	5.5283
	(0.6346)	(0.4376)		
No. of trades	197.9300	206.5100	524	274
	(21.3524)	(12.9244)		
Win%	54.9958	52.6284	53.6260	53.6496
	(2.2384)	(1.1910)		
Profitable stock count	35.1300	32.2300	32	41
	(2.1113)	(1.4897)		
Non-profitable stock count	16.8700	19.7700	20	11
	(2.1113)	(1.4897)		
Profit ratio	67.5577	61.9808	61.5385	78.8462
	(4.0602)	(2.8649)		
Profitable stock net profit	13.4125	11.4923	15.5169	12.0899
(millions)	(2.3186)	(0.8384)		
Non-profitable stock net profit	-0.3852	-0.4315	-0.4769	-0.2250
(millions)	(0.0716)	(0.0328)		
Stock net profit	13.0272	11.0608	15.0400	11.8649
(millions)	(2.3658)	(0.8350)		

Table 9: Performance of experiment 1 for PRS, WS and the two best component rules in the testing period for transaction cost C = 0.005. PRS and WS are run for 100 times.

	PRS	WS	Best MA (150-125)	Best TRB (125)
ANP (%)	16.4227	15.1676	17.8106	15.6492
	(1.8136)	(0.7750)		
Sharpe ratio	0.9458	0.9258	1.0472	1.0842
	(0.0429)	(0.0204)		
Payoff ratio	6.1173	5.9102	3.7255	5.3807
	(0.6158)	(0.4586)		
No. of trades	191.3300	206.7900	524	274
	(20.0716)	(12.7099)		
Win%	54.2178	51.7759	51.5267	52.9197
	(2.2011)	(1.1569)		
Profitable stock count	34.8500	31.3400	32	39
	(2.0369)	(1.5905)		
Non-profitable stock count	17.1500	20.6600	20	13
	(2.0369)	(1.5905)		
Profit ratio	67.0192	60.2692	61.5385	75.0000
	(3.9171)	(3.0587)		
Profitable stock net profit	12.8943	11.3973	14.6769	11.7012
(millions)	(2.3736)	(0.8687)		
Non-profitable stock net profit	-0.4194	-0.4844	-0.5811	-0.2621
(millions)	(0.0679)	(0.0336)		
Stock net profit	12.4748	10.9129	14.0958	11.4391
(millions)	(2.4077)	(0.8611)		

7. RESULTS OF EXPERIMENT 2

Table 10, 11, 12 show performance of experiment 2 for PRS, WS and the seven best component rules in the testing period for transaction cost C = 0.001, 0.002, 0.005 respectively. We now consider the case of transaction cost C = 0.001. PRS's average ANP is greater than all the best component rules' ANPs (except the best RSI's ANP) and WS's average ANP. While we run the program in HPC (high performance computing facility), sometime HPC is busy and so we adjust program submission configuration as the number of core to be 4 and select queue as short which has maximum running time of 24 hours. This configuration of program submission is to increase the chance of HPC selecting our program to run first. But the downside of this method is there is only 4 cores available and the running time is limited to 24 hours. However, some of the runs of our program for experiment 2 require more than 24 hours and these runs are terminated by HPC. When the server is free, we run our programs until its completion. The above fact biases our results of experiment 2. That is the results are geared towards runs of program which terminate early. PRS's average ANP of transaction cost C = 0.002 is similar to the case of transaction cost C = 0.001. We postulate that if we can let all our programs in experiment 2 run to its completion, PRS's average ANP of transaction cost C = 0.002 may be less than PRS's average ANP of transaction C = 0.001. PRS's average ANP of transaction cost C = 0.005 is less than PRS's average ANP of transaction cost C = 0.001 and 0.002.

Let set1 = set of all component rules in experiment 1 and set2 = set of all component rules in experiment 2. For C = 0.001, PRS's average ANP in experiment 2 is greater than PRS's average ANP of experiment 1. This may be due to the fact that set1 is subset of set2 and the best ANP of all rules in set2 which comes from the best RSI is greater than the best ANP of all rules in set1 which comes from the best MA.

Table 10: Performance of experiment 2 for PRS, WS and the seven best component rules in the testing period for transaction cost C = 0.001. PRS and WS are run for 100 times.

	PRS	WS	Best MA (150-125)	Best TRB (125)	Best BB (10-1.9)	Best RSI (19-80-20)	Best STO (10-7-90-20)	Best MACD (100-20-9)	Best OBVA (75-50)
ANP (%)	19.3833	17.2360	18.7529	16.1374	11.3149	20.9548	11.2678	15.9618	10.4040
	(1.1688)	(2.1521)							
Sharpe ratio	0.9171	0.9684	1.0948	1.1138	1.0474	0.9398	0.7709	0.9361	0.8095
	(0.0294)	(0.0539)							
Payoff ratio	4.4648	1.4857	3.5056	5.4719	0.6984	5.7671	1.6144	3.7563	1.3686
	(2.2547)	(0.3531)							
No. of trades	505.6500	1,875.8700	524	274	2769	343	259	1350	2215
	(215.5600)	(562.1024)							
Win%	55.4643	57.7254	54.9618	54.3796	70.6031	48.1050	72.9730	35.2593	54.9887
	(8.1980)	(2.0000)							
Profitable stock count	42.2300	38.5200	34	41	44	45	44	32	43
	(2.0590)	(1.7551)							
Non-profitable stock count	9.7700	13.4800	18	11	8	7	8	20	9
	(2.0590)	(1.7551)							
Profit ratio	81.2115	74.0769	65.3846	78.8462	84.6154	86.5385	84.6154	61.5385	82.6923
	(3.9596)	(3.3752)							
Profitable stock net profit	16.5682	13.8650	15.8071	12.2233	7.1926	18.8077	7.1682	12.2862	6.5711
(millions)	(1.6330)	(2.8801)							
Non-profitable stock net profit	-0.2563	-0.3308	-0.4416	-0.2139	-0.1347	-0.1862	-0.1516	-0.4839	-0.2929
(millions)	(0.0635)	(0.0405)							
Stock net profit	16.3119	13.5342	15.3655	12.0094	7.0579	18.6215	7.0165	11.8023	6.2781
(millions)	(1.6649)	(2.8939)							

Table 11: Performance of experiment 2 for PRS, WS and the seven best component rules in the testing period for transaction cost C = 0.002. PRS and WS are run for 100 times.

	PRS	WS	Best MA (150-125)	Best TRB (125)	Best BB (10-2.1)	Best RSI (19-80-20)	Best STO (10-7-90-20)	Best MACD (100-20-9)	Best OBVA (150-100)
ANP (%)	19.4247	17.0066	18.5163	16.0151	9.8830	20.7901	11.1282	15.3327	9.4853
	(1.3150)	(2.6297)							
Sharpe ratio	0.9121	0.9407	1.0829	1.1064	0.9600	0.9330	0.7632	0.9010	0.7096
	(0.0349)	(0.0652)							
Payoff ratio	5.5221	1.6490	3.6293	5.5283	0.7008	5.8135	1.5867	3.7042	1.6271
	(3.4159)	(0.4563)							
No. of trades	355.9600	1,582.5100	524	274	2457	343	259	1350	1180
	(169.1354)	(479.8948)							
Win%	58.1379	56.5227	53.6260	53.6496	69.9634	47.5219	72.9730	34.8148	52.0339
	(8.0154)	(2.2287)							
Profitable stock count	42.3100	36.9200	32	41	44	45	44	30	27
	(2.1541)	(2.1114)							
Non-profitable stock count	9.6900	15.0800	20	11	8	7	8	22	25
	(2.1541)	(2.1114)							
Profit ratio	81.3654	71.0000	61.5385	78.8462	84.6154	86.5385	84.6154	57.6923	51.9231
	(4.1426)	(4.0605)							
Profitable stock net profit	16.6379	13.7191	15.5169	12.0899	6.0576	18.5574	7.0525	11.6541	6.2831
(millions)	(1.8595)	(3.3864)							
Non-profitable stock net profit	-0.2509	-0.3909	-0.4769	-0.2250	-0.2057	-0.1941	-0.1581	-0.5758	-0.7471
(millions)	(0.0701)	(0.0591)							
Stock net profit	16.3870	13.3282	15.0400	11.8649	5.8519	18.3633	6.8945	11.0783	5.5360
(millions)	(1.8900)	(3.4252)							

Table 12: Performance of experiment 2 for PRS, WS and the seven best component rules in the testing period for transaction cost C = 0.005. PRS and WS are run for 100 times.

	PRS	WS	Best MA (150-125)	Best TRB (125)	Best BB (35-2.2)	Best RSI (19-80-20)	Best STO (10-7-90-20)	Best MACD (100-20-9)	Best OBVA (200-125)
ANP (%)	18.7315	19.7570	17.8106	15.6492	6.2707	20.3000	10.7114	13.4702	7.7571
	(1.4269)	(2.5804)							
Sharpe ratio	0.8929	0.9191	1.0472	1.0842	0.6274	0.9126	0.7404	0.7981	0.6761
	(0.0474)	(0.0535)							
Payoff ratio	5.7725	3.2659	3.7255	5.3807	0.7912	5.9559	1.5662	3.5439	2.0496
	(3.1138)	(2.2052)							
No. of trades	254.2200	518.0600	524	274	742	343	259	1350	929
	(143.0819)	(360.6011)							
Win%	63.4532	55.8937	51.5267	52.9197	70.0809	45.7726	72.2008	33.6296	48.1163
	(9.9124)	(5.3410)							
Profitable stock count	41.5800	40.0100	32	39	41	43	44	26	28
	(3.1883)	(3.6055)							
Non-profitable stock count	10.4200	11.9900	20	13	11	9	8	26	24
	(3.1883)	(3.6055)							
Profit ratio	79.9615	76.9423	61.5385	75.0000	78.8462	82.6923	84.6154	50.0000	53.8462
	(6.1313)	(6.9337)							
Profitable stock net profit	15.6931	17.3697	14.6769	11.7012	3.5077	17.8331	6.7132	9.9565	4.9004
(millions)	(1.9556)	(3.3694)							
Non-profitable stock net profit	-0.2744	-0.3040	-0.5811	-0.2621	-0.2489	-0.2240	-0.1768	-0.8660	-0.6476
(millions)	(0.1112)	(0.1189)							
Stock net profit	15.4187	17.0657	14.0958	11.4391	3.2587	17.6092	6.5363	9.0905	4.2528
(millions)	(2.0047)	(3.4556)							

8. CONCLUSION

In this study, we conduct 100 rerun of each experiment in Wang et al, 2014a and b to increase reliability of performance measurement. We conclude that in terms of average ANP of PRS and ANP of component rules, PRS in each experiment of Wang 2014a and b does not outperform all component rules.

9. ACKNOWLEDGEMENT

We would like to thank Prof. Sira Suchintabandid and Prof. Peerapong Dhangwatnotai for their kind support of this study. The authors acknowledge National e-Science Infrastructure Consortium for providing computing resources that have contributed to the research results reported within this paper (URL:http://www.e-science.in.th).

10. REFERENCES

Wang, Fei, L. H. Philip, and David W. Cheung. *“Combining technical trading rules using particle swarm optimization."* Expert Systems with Applications 41.6 (2014): 3016-3026.

Wang, Fei, Philip LH Yu, and David Wai-lok Cheung. *"Combining technical trading rules using parallel particle swarm optimization based on Hadoop."* Neural Networks (IJCNN), 2014 International Joint Conference on. IEEE, 2014.

Ratnaweera, Asanga, Saman K. Halgamuge, and Harry C. Watson. *"Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients."* Evolutionary Computation, IEEE Transactions on 8.3 (2004): 240-255.