POLYNOMIAL TIME HEURISTIC OPTIMIZATION METHODS APPLIED TO PROBLEMS IN COMPUTATIONAL FINANCE

Thesis booklet for Ph.D. dissertation

Fogarasi Norbert

Supervisor:
Dr. Levendovszky János, D. Sc.
Doctor of the Hungarian Academy of Sciences

Department of Telecommunications
Budapest University of Technology and Economics

Budapest, 2014

1. Introduction and scope of the theses

Computational finance is a branch of applied computer science that deals with problems of practical interest in finance. It is a relatively new discipline whose birth can be traced back to the work of Harry Markowitz in the 1950s and since then has provided many problems of deep algorithmic interest. Presently, it encompasses various new numerical methods in the fields of optimization, statistics, and stochastic processes developed for different financial applications.

Markowitz conceived of the portfolio selection problem as an exercise in mean-variance optimization. His key finding was that diversification, as a means of reducing the variance and therefore increasing the predictability of an investment portfolio is of critical importance, even at the cost of reducing expected return [32]. For his groundbreaking work, Markowitz received the Nobel Prize in 1990. There are many other problems in finance which have been raised since the 1950's such as algorithmic trading, the validation of efficient markets hypothesis, option pricing, financial time series analysis and prediction.

The area of algorithmic trading, the use of electronic platforms for entering trading orders with an algorithm deciding on aspects of the order such as timing, price or quantity, has attracted a lot of attention over the last decade. In particular, high frequency trading has gained a lot of ground in international markets, in which computers make algorithmic decisions on high frequency, often sub-second, tick-by-tick data, before human traders are capable of processing the information they observe. It is estimated that as of 2009, high frequency trading algorithms are responsible for over 70% of all equity trading volume in the US [22], so the area has also attracted a lot of academic attention in recent years. At the same time, parallel computational techniques on hardware, such as GPGPU and FPGA have also evolved, so traditional methods which could historically only been used in low or medium frequency trading can now be applied in real-time settings. On the other hand, many of the problems which arise during the development of optimal trading algorithms are of high complexity which makes the application of fast approximation methods necessary for practical use in high frequency trading.

Another developing area of computational finance is the increased use of Monte Carlo techniques for simulation of market processes under certain stochastic model assumptions. This requires the use of an increasing amount of computational resources and gives rise to problems of optimal usage of the resources via scheduling. Again, many of the combinatorial problems of optimal scheduling have been proven to be of exponential or NP hard complexity which have made the use of polynomial approximation techniques necessary.

As demonstrated above, many of the fundamental problems which have arisen in the field have been shown to be of exponential complexity or NP hard. As a result, the fundamental objective of computational finance is to find numerical algorithms yielding approximate solutions within practically viable timeframes to these problems.

The present dissertation has the same aim for two specific problems:

1) the problem of finding sparse portfolios, based on historical data, which maximize predictability (or mean reversion); and
2) the problem of finding optimal schedules for a set of identical computational resources for jobs (Monte Carlo simulations) with prescribed sizes, weights and deadlines which minimize total weighted tardiness.

Even though these two problems are seemingly rooted in different areas they are brought together in the area of computational finance (portfolio optimization and optimal resource scheduling), and the theme of finding fast, heuristic approximate solutions to NP hard problems with applications in finance, is common. As a result, the theses provide novel and important contributions to the field of computational finance which have very practical applications, as will be demonstrated.

1.1 Technological motivations and existing results

In this section, an outline of the motivations and practical applications of the selected problems is given, as well as a brief presentation of the existing research results.

1.1.1 Sparse, mean reverting portfolio selection

Ever since the seminal paper of Markowitz [32], selecting portfolios which are optimal in terms of risk-adjusted returns has been an active area of research both by academics and financial practitioners. At the same time, mean reversion, as a classic indicator of predictability, has also received a lot of attention over the last few decades. It has been shown that equity excess returns over long horizons are mean-reverting and therefore contain an element of predictability [21,31,36]. While there exist simple and reliable methods to identify mean reversion in univariate time series, selecting portfolios from multivariate data which exhibit this property is a much more difficult problem. This can be approached by the Box-Tiao procedure [15] to extract cointegrated vectors by solving a generalized eigenvalue problem.

In his recently published article, d'Aspremont in [18] posed the problem of finding mean-reverting portfolios which are sparse. The practical application of this is the possibility to develop profitable *convergence trading* strategies based on buying the portfolio when it is below the long-term mean and selling when the portfolio has appreciated over this level. Sparseness of the portfolio is desirable for reducing transaction costs associated with convergence trading as well as for increasing the interpretability of the resulting portfolio. He developed a new approach to solve the problem by using semidefinite relaxation and compared the efficiency of this solution to the simple greedy algorithm in a number of markets. d'Aspremont makes the assumption that the underlying processes follow a first order vector autoregressive VAR(1) process and uses historical data to estimate the model parameters. There is vast literature on the topic of parameter estimation of VAR(1) processes, recent research has focused on sparse and regularized covariance estimation [10,17,38].

In the first thesis group, I investigated new, dense estimation methods for determining the covariance matrix of an observed VAR(1) process. I also examine methods to determine the parameters of mean-reverting processes based on the Ornstein-Uhlenbeck process. Finally, I investigate new heuristic methods for determining the optimal, most predictable portfolio under cardinality constraints.

1.1.2 Optimal scheduling on identical machines

Scheduling theory has a special application in running large scale Monte Carlo simulations in financial services firms for evaluating risks and pricing. As a result, finding optimal schedules in real time which minimize the completion time of jobs subject to capacity constraints is an especially important task. More specifically, in the area of computational finance, the problems of portfolio selection, pricing and hedging of complex financial instruments requires an enormous amount of computational resources whose optimal usage is of utmost importance to investment banks. The prices and risk sensitivity measures of complex portfolios need to be reevaluated daily, for which an overnight batch of calculations is scheduled and performed for millions of financial transactions, utilizing thousands of computing nodes. Each job has a well-defined priority and required completion time for availability of the resulting figures to the trading desk, risk managers and regulators. The jobs can generally be stopped and resumed at a later point on a different machine which is referred to as *preemption* in scheduling theory. For simplicity of modeling the problem, machines are generally assumed to be identical and there is a known, constant number of machines available. Tardiness of an individual job under a given schedule is defined as the amount of time by which the job finishes after its prescribed deadline, and is considered to be zero if the job finishes on or before the deadline.

The problem of finding optimal schedules for jobs running on identical machines has been extensively studied over the last three decades. In his paper, Sahni [40] presents an *O(n log mn)* algorithm to construct a *feasible* schedule, one that meets all due times, if one exists, for *n* jobs and *m* machines. The basic idea of the algorithm is to schedule jobs with earliest due dates first, but fill up machines with smaller jobs if possible. Note that this method allows the development of an algorithm to compute the minimal amount of unit capacity for which a feasible schedule exists. This result has been extended to machines with identical functionality but different processing speed, termed *uniform machines*, and jobs with both starting times and deadlines; Martel [33] constructs a polynomial time feasible schedule for this problem, if one exists. However, the scheduling task becomes more difficult when a feasible schedule does not exist and the goal is to minimize some measure of delinquency, often termed *tardiness*.

In case of minimizing the maximum tardiness across all jobs, Lawler [27] shows that the problem is solvable in polynomial time, even with some precedence constraints. Martel [33] also used his construction to create a polynomial time algorithm to find the schedule which minimizes maximum lateness. However, if the measure concerns the total tardiness instead of the maximal one, then even the single machine, total tardiness problem (without weights) was proven to be NP-hard by Du and Leung [19]. A pseudopolynomial algorithm has been developed by Lawler [26] for this problem, using dynamic programming, but this is for the 1-machine problem and does not have good practical runtime characteristics.

Once the NP-hardness of the TWT problem was established, most of the research work on the problem concerned the development of fast, heuristic algorithms. Dogramaci and Surkis [20] propose a simple heuristic for the total (non-weighted) tardiness problem without preemption. Rachamadugu and Morton [37] then studied the identical machine, total weighted tardiness problem without preemption. They proposed a myopic heuristic and compared this to earliest due date (EDD), weighted shortest processing time (WSPT) and Montagne's rule on small problem sizes (2 or 5 jobs in total). Azizoglu and Kirca [12] worked on an algorithm to find optimal schedule for the unweighted total tardiness problem without preemption, but their branch and bound exponential algorithm is too slow, in practice, for problems with more than 15 jobs. Armentano and Yamashita [11] examined the non-weighted problem without preemption, and starting from the KPM heuristic of Koulamas [25] improved upon it, using tabu search. Guinet [23] applies simulated annealing to solve the problem with uniform and identical machines and a lower bound is presented in order to evaluate the performance of the proposed method. More recently, Sen et al. [41] surveyed the existing heuristic algorithms for the single-machine total tardiness and total weighted tardiness problems while Biskup et al. [14] did this for the identical machines total tardiness problem and also proposed a new heuristic. Akyol and Bayhan [10] provide an excellent recent review of artificial neural network based approaches to scheduling problems and proposes a coupled gradient network to solve the weighted earliness plus tardiness problem on multiple machines. The feasibility of the method is illustrated on a single 8-job scheduling problem.

I designed a novel heuristic for the TWT problem, based on the Hopfield Neural Network approach which is shown to perform better than existing simple heuristics and has desirable scaling characteristics. Maheswaran et al. [28] applied a similar approach to the single machine TWT problem and his results were encouraging for a specific 10-job problem.

In the second thesis group, I investigate novel, polynomial time heuristics for the solution of the NP hard TWT problem. I further investigate opportunities to improve upon the solution by the use of random perturbations. Finally, I test the newly developed methods on a large number of randomly generated problems as well as on real scheduling data set obtained from Morgan Stanley, one of the largest financial institutions in the world.

2. Models and methods used in the research

In order to achieve the results presented in the dissertation, a number of models and computational methods have been used and developed. These are outlined in this chapter.

For solving the problem of sparse, optimally mean reverting portfolios, I follow the construction outlined in [18]. D'Aspremont made a number of assumptions in the construction: he assumed the underlying processes follow a VAR(1) model and that the resulting mean reverting portfolios can be described by the Ornstein-Uhlenbeck equation. For the estimation of model parameters, I used the Moore-Penrose pseudoinverse to solve the equation arising during the maximum likelihood estimation of the recursion matrix and I developed a novel recursive method for estimating the covariance matrix. Having estimated the parameters of the VAR(1) model, I mapped the prediction maximization problem to a generalized eigenvalue problem with cardinality constraint which has been shown to be NP hard [34]. Therefore, I applied simulated annealing to the problem, in order to find a polynomial time approximation and I compared this to a number of benchmark methods. These include exhaustive search, greedy search and a novel heuristic which I developed based on truncation. Having found the portfolio, I developed a novel method to find its long-term mean using pattern matching. Finally, the economic viability of the method was verified by running numerical simulations based on a novel convergence trading strategy which I developed based on a decision theoretic formulation.

For solving the problem of finding optimal schedules on identical machines which minimize the total weighted tardiness of jobs, I used a binary scheduling matrix model to represent the schedules. Furthermore, I studied a number of existing heuristic methods (EDD, WSPT, LWPF, LBS) which I subsequently used as benchmarks for evaluating the performance of my novel approach. I used matrix algebraic transformations to convert the problem to quadratic form and built the constraints into the objective function, using heuristic constants. This allowed me to use Lyapunov convergence via the Hopfield neural network to find polynomial time approximate solutions to this NP hard problem. I further improved upon this method by introducing random perturbation to the intelligently selected initial point.

All model and method implementations were performed in the MATLAB computational environment, where numerical simulations were run on both synthetic and real, historical data for both problems at hand. The results presented in the dissertation are based on the numerical simulations performed in this way for both problems.

Models used	VAR(1) model Ornstein-Uhlenbeck model	Binary scheduling matrix
Methods applied	Maximum likelihood estimation Moore-Penrose pseudoinverse Novel recursive covariance matrix estimation Benchmark methods (Exhaustive search, greedy search, novel truncation method) Simulated annealing Novel long-term mean estimation based on pattern matching Novel convergence trading based on decision theoretic formulation	Heuristic optimization (EDD, WSPT, LWPF, LBS) benchmark methods Combinatorial optimization of quadratic forms Lyapunov convergence Hopfield neural network Random perturbations
Validation	Numerical simulations on synthetic and real historical data in MATLAB	
Results	Thesis group 1	Thesis group 2

Figure 1. Models, methods and validation used to derive the results of this dissertation.

3. New scientific results

I. Thesis group: I developed a new method for estimating parameters of the VAR(1) process which also provides a goodness of fit measure. Furthermore, I introduce a novel method for estimating the long-term mean of an Ornstein-Uhlenbeck process based on pattern matching. I also develop a new method for portfolio selection, adapting the simulated annealing method and finally develop a simple convergence trading algorithm through which I show the practical viability of my methods.

(Publications connected to this thesis group: 1, 2, 5, 6)

Following the construction of d'Aspremont [18], I view the asset prices as a stationary, first order, vector autoregressive VAR(1) process. Let $s_{i,t}$ denote the price of asset *i* at time instant *t*, where $i = 1,...,n$ and *t* are positive integers and assume that $\mathbf{s}_t^T = (s_{1,t},...,s_{n,t})$ is subject to a first order vector autoregressive process, VAR(1), defined as follows:

$$\mathbf{s}_t = \mathbf{A}\mathbf{s}_{t-1} + \mathbf{W}_t, \tag{1}$$

where **A** is an *nxn* matrix and $\mathbf{W}_t \sim N(0,\sigma I)$ are i.i.d. noise terms for some $\sigma > 0$. Let **G** denote the stationary covariance matrix of process $\mathbf{s}_t$. I assume that historical time series of $\mathbf{s}_t$ can be observed and the task is to estimate matrices **A** and **G.**

I first observe that if the number of assets *n* is greater than or equal to *m*, the length of the observed time series, then **A** can be estimated by simply solving exactly the linear system of equations:

$$\hat{\mathbf{A}}\mathbf{s}_{t-1} = \mathbf{s}_t. \tag{2}$$

Note that if $n > m$, this system is underdetermined, so there are infinitely many solutions. In this case, I considered the subsystem consisting of the first *m* assets to obtain a unique solution. This gives a perfect VAR(1) fit for the time series for cases where I have a large portfolio of potential assets in relation to the amount of data observed (e.g. considering daily close prices over a 1-month period of all 500 stocks which make up the S&P 500 index), from which a sparse mean-reverting portfolio is to be chosen.

In most of the applications, however, the length of the available historical time series is greater than the number of assets considered, so (2) is overdetermined, and **A** is estimated using, for example, least squares estimation techniques, as in

$$\hat{\mathbf{A}} = \arg\min_{\mathbf{A}} \sum_{t=2}^{m} \left\| \mathbf{s}_t - \mathbf{A}\mathbf{s}_{t-1} \right\|^2 \tag{3}$$

where $\|\cdot\|$ denotes the Euclidian norm.

Equating to zero the partial derivatives of the above expression with respect to each element of the matrix **A**, I obtain the following system of equations:

$$\sum_{k=1}^{n}\hat{\mathbf{A}}_{i,k}\sum_{t=2}^{m}s_{k,t-1}s_{j,t-1}=\sum_{t=2}^{m}s_{i,t}s_{j,t-1}\forall i,j=1,...,n. \tag{4}$$

Solving for $\hat{\mathbf{A}}$ and switching back to vector notation for **s**, I obtain

$$\hat{\mathbf{A}}=\sum_{t=2}^{m}\mathbf{s}_t\mathbf{s}_{t-1}^T\left(\sum_{t=2}^{m}\mathbf{s}_{t-1}\mathbf{s}_{t-1}^T\right)^+, \tag{5}$$

where $\mathbf{M}^+$ denotes the Moore-Penrose pseudoinverse of matrix **M**. Note that the Moore-Penrose pseudoinverse is preferred to regular matrix inversion, in order to avoid problems with the singularity of $\mathbf{s}_{t-1}\mathbf{s}_{t-1}^T$.

Assuming that the noise terms in equation (1) are i.i.d. with $\mathbf{W}_t \sim N(0,\sigma I)$ for some $\sigma > 0$, I obtain the following estimate for σ using $\hat{\mathbf{A}}$ from (5):

$$\hat{\sigma}=\sqrt{\frac{1}{n(m-1)}\sum_{t=2}^{m}\left\|\mathbf{s}_t-\hat{\mathbf{A}}\mathbf{s}_{t-1}\right\|^2}. \tag{6}$$

In the more general case that the terms of $\mathbf{W}_t$ are correlated, I can estimate the covariance matrix **K,** of the noise as follows:

$$\hat{\mathbf{K}}=\frac{1}{m-1}\sum_{t=2}^{m}(\mathbf{s}_t-\hat{\mathbf{A}}\mathbf{s}_{t-1})\ (\mathbf{s}_t-\hat{\mathbf{A}}\mathbf{s}_{t-1})^T. \tag{7}$$

This noise covariance estimate is used below in the estimation of the covariance matrix.

I.1. I have developed a new numerical method for estimating the covariance of historical time series, assuming a VAR(1) model, which yields fast and reliable estimation as well as a goodness of fit measure.

This method is based on the following formula:

$$\mathbf{G}(k+1)=\mathbf{G}(k)-\delta(\mathbf{G}(k)-\mathbf{A}\mathbf{G}(k)\mathbf{A}^T-\mathbf{K}), \tag{8}$$

where δ is a constant between 0 and 1, $\mathbf{G}(k)$ is the covariance matrix estimate on iteration *k*. Provided that the starting point for the numerical method, **G**(0), is positive definite (e.g. the sample covariance matrix) and since the estimate of **K** is positive definite, by construction, this iterative method produces an estimate which is positive definite. It provides a smoother convergence to the true covariance than the sample covariance estimate, as can be seen in the below figure which was produced using synthetic data.

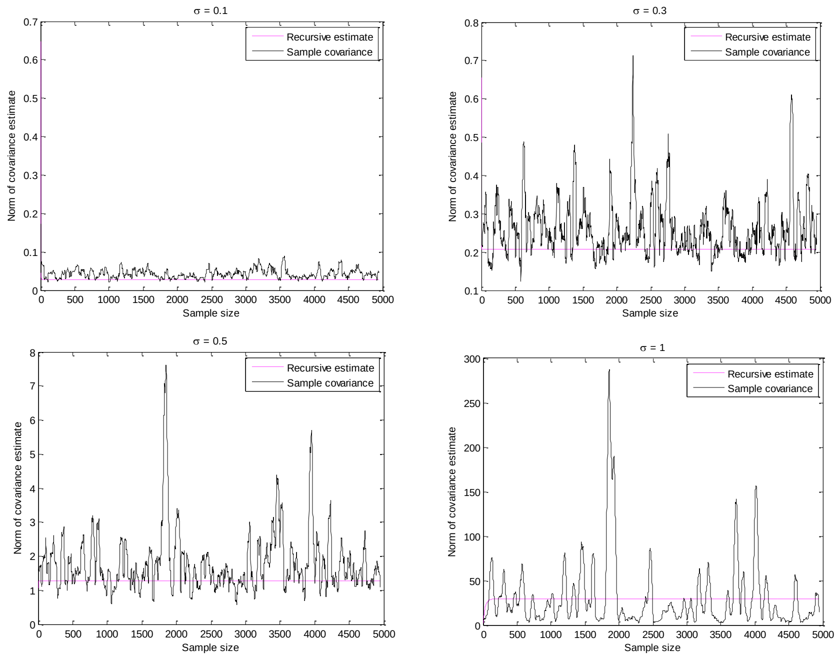

Figure 2: Sample covariance and recursive covariance estimates over sliding windows of size 50 over 5000 samples for σ =0.1, 0.3, 0.5, 1 (note the differences in scaling of the plots)

This method also provides a goodness of model fit measure when the above recursive estimate is compared to the sample covariance. This can be used during convergence trading to set the level of confidence in having obtained a truly mean-reverting portfolio.

My next result concerns the estimation of the parameters of the Ornstein-Uhlenbeck equation, given by

$$dp_t = \lambda\left(\mu - p_t\right)dt + \sigma dW_t, \tag{9}$$

where W_t is a Wiener process and $\lambda > 0$ (mean reversion coefficient), μ (long-term mean), and $\sigma > 0$ (portfolio volatility) are constants.

I.2. I have introduced a novel method for mean estimation of Ornstein-Uhlenbeck processes based on pattern matching. In this way, the most likely mean is selected by maximizing the underlying Gaussian densities of the OU processes. The new estimation is given by the following expression

$$\hat{\mu}_3 := \frac{\sum_{i=1}^{t}\sum_{j=1}^{t}\left(\mathbf{U}^{-1}\right)_{i,j}\left[\mu(0)\ \left(2e^{-\lambda(i+j)} - e^{-\lambda i} - e^{-\lambda j}\right) - 2\mathbf{p}_j\left(e^{-\lambda i} - 1\right)\right]}{\sum_{i=1}^{t}\sum_{j=1}^{t} 2\left(\mathbf{U}^{-1}\right)_{i,j}\left(e^{-\lambda(i+j)} - e^{-\lambda i} - e^{-\lambda j} + 1\right)}, \qquad (10)$$

where $\mathbf{U}_{ij} := E\left(p(t-i)\,p(t-j)\right) = \frac{\sigma^2}{2\lambda} e^{-\lambda(i+j)}\left(e^{2\lambda i} - 1\right)$ is the time-correlation matrix of $\mathbf{p}_t$.

By this method a more stable than linear regression and more accurate than sample mean can be obtained as shown in the following figures.

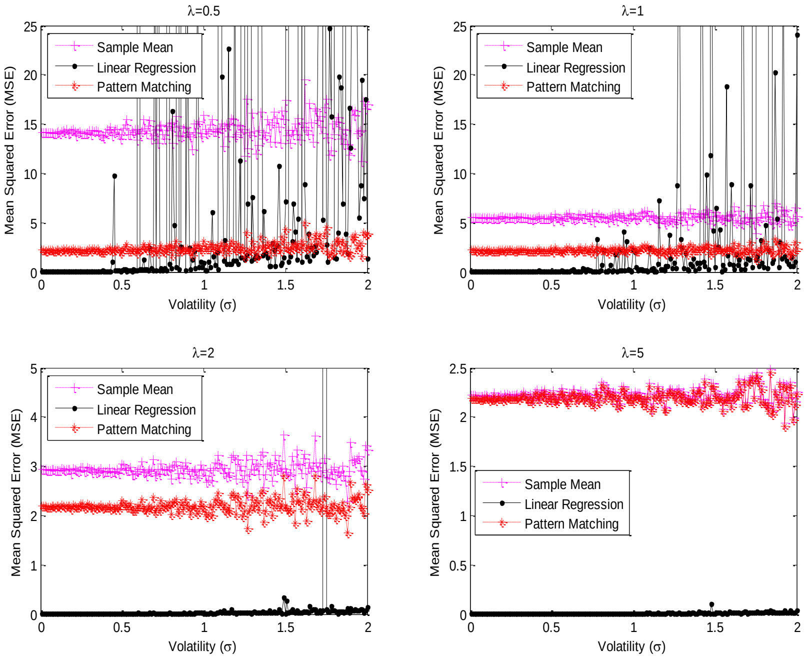

Figure 3: MSE of each μ estimate as a function of σ. λ =0.5, 1, 2, 5 , $\mu_0 = 0$ and sample size of 20.

The main task after identifying the mean reverting portfolio and obtaining an estimate for its long-term mean μ, is to verify whether $\mu(t) < \mu$ or $\mu(t) \geq \mu$ based on observing the samples $\left\{ p(t) = \mathbf{x}^T \mathbf{s}(t), t = 1,...,T \right\}$. This verification can be perceived as a decision theoretic problem, since direct observations of $\mu(t)$ are not available.

If process $p(t)$ is in stationary state then the samples $\left\{ p(t), t = 1,...,T \right\}$ are generated by a Gaussian distribution $N\left(\mu, \sqrt{\frac{\sigma^2}{2\lambda}} \right)$. As a result, for a given rate of acceptable error ε, I can select an α for which

$$\int_{\mu-\alpha}^{\mu+\alpha} \frac{1}{\sqrt{2\pi\sigma^2/2\lambda}} e^{-\frac{(u-\mu)^2}{\sigma^2/\lambda}} du = 1-\varepsilon. \tag{11}$$

I.3. I have developed a simple trading strategy based on the decision theoretic formulation of trading. This results in a convergence trading algorithm which can be used to compare the profits generated by various portfolio selection algorithms.

The repeating steps of the algorithm in each time interval are summarized in the following figure:

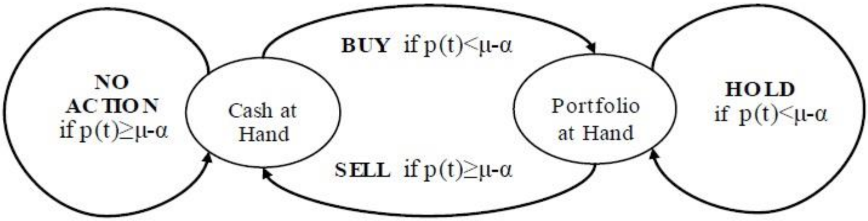

Figure 4: Flowchart for simple convergence trading of mean reverting portfolios

I.4. I adopted the simulated annealing optimization method to the problem of maximizing mean reversion under cardinality constraints. I developed convergence rules and neighborhood functions which perform well on a range of randomly generated examples. This method consistently outperformed the greedy heuristic in 20% of the randomly generated cases within acceptable runtimes.

In some synthetic cases, simulated annealing outperforms the greedy method, in terms of producing consistently higher mean reversion coefficients for various levels of cardinality constraints. An example of this is shown in the following figure.

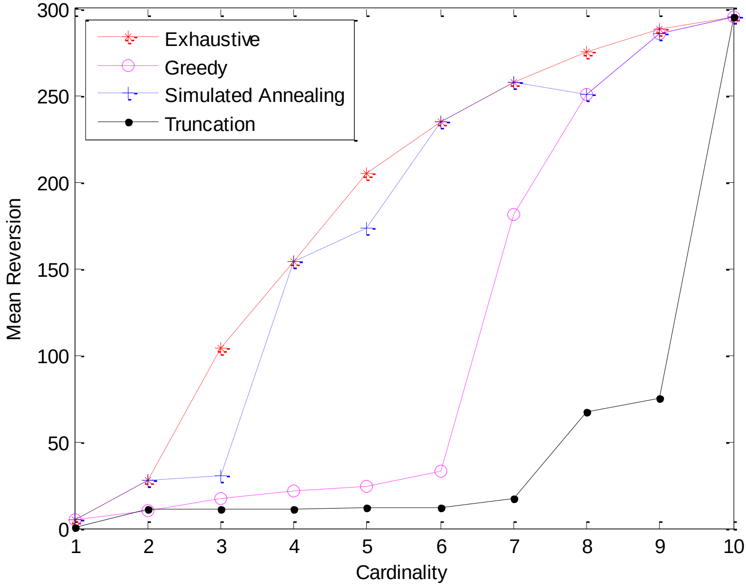

Figure 5: Comparison of portfolio selection methods of various cardinality on n=10-dimensional generated VAR(1) data.

As a proof of concept, I have made a complete end-to-end implementation of the parameter estimation techniques, the improved portfolio selection algorithm utilizing simulated annealing and the decision theoretic trading strategy. I have found that on real historical data, pipelining these novel methods produce meaningful excess returns.

II. Thesis group: I have shown how the TWT problem can be converted to quadratic form and describe a specific quadratic objective function which allows the use of the Hopfield neural network to solve the TWT problem. I show how this method outperforms simpler heuristics on a large number of randomly generated problems and I further improve this method by introducing intelligent starting points for the iterative process and then random perturbations to it. I show the practical applicability of these methods to a specific overnight scheduling problem arising at Morgan Stanley, one of the largest financial institutions in the world.

(Publications connected to this thesis group: 3, 4)

I worked on the total weighted tardiness (TWT) problem in the following formalism. Given N jobs with sizes $\mathbf{x} = \{x_1, x_2, x_3, \ldots x_N\} \in \mathbb{N}^N$, the processing of the jobs can be stopped and resumed arbitrary at any time, so the processing time units of each job need not be contiguous. In the literature this condition is known as *preemption* and also assumes a task started on one machine can continue on another [16]. For each job a cutoff time is prescribed by $\mathbf{K} = \{K_1, K_2, K_3, \ldots, K_N\} \in \mathbb{N}^N$. This constraint defines the time within which the job is to be completed. The constant number of processors, the capacity of the system is denoted by *V*. Also given is a vector $\mathbf{w} = \{w_1, w_2, w_3, \ldots, w_N\} \in \mathbb{R}^N$, $w_i \geq 0, \forall i = 1, \ldots, N$ denoting the relative priority (or weight) of each job, which can be used in the definition of the objective function. A schedule is represented by a binary matrix $\mathbf{C} \in \{0,1\}^{N \times L}$ where $C_{i,j} = 1$ if job i is being processed at time slot j, and L denotes the length of the schedule. An example is given in (12), where the parameters are the following: $V = 2$, $N = 3$, $\mathbf{x} = \{2,3,1\}$, $\mathbf{K} = \{3,3,3\}$.

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix} \tag{12}$$

In order to evaluate the effectiveness of a given schedule **C**, *tardiness* of a job is defined as follows:

$$T_i = \max\left(0, F_i - K_i\right), \tag{13}$$

where F_i is the actual finish time of job i under schedule **C**: $F_i = \arg\max_j \left\{C_{i,j} = 1\right\}$ (The position of the last 1 in the i^{th} row in scheduling matrix $\mathbf{C}$.)

The problem which needs to be solved can now be stated formally as follows:

$$\mathbf{C}_{opt} := \arg\min_{\mathbf{C}} \sum_{i=1}^{N} w_i T_i \tag{14}$$

Under the following constraints:

- The length of jobs in the scheduling matrix equals the pre-defined amounts

$$\sum_{j=1}^{L} C_{i,j} = x_i, \forall i = 1,...,N \tag{15}$$

- The number of scheduled jobs at a given time instant does not exceed the capacity of the system:

$$\sum_{i=1}^{N} C_{i.j} \le V, \forall j = 1,...,L \tag{16}$$

II.1. I have proven that the TWT problem can be converted to a quadratic optimization problem, including the constraints with heuristic coefficients.

I have converted the full optimization problem into quadratic form,

$$f\left(\mathbf{y}\right) = -\frac{1}{2}\mathbf{y}^T\mathbf{W}\mathbf{y} + \mathbf{b}^T\mathbf{y}, \tag{17}$$

with

$$\mathbf{W} = \alpha \mathbf{W}_A + \beta \mathbf{W}_B + \gamma \mathbf{W}_C \in \mathbb{R}^{NL \times NL} \quad (18)$$

and

$$\mathbf{b} = \alpha \mathbf{b}_A + \beta \mathbf{b}_B + \gamma \mathbf{b}_C \in \mathbb{R}^{NL \times 1}, \quad (19)$$

where the A subscript corresponds to the objective function, the B and C subscripts correspond to the two constraints and α, β, γ are heuristic coefficients.

The minimization problem is equivalent to:

$$\mathbf{C}_{opt} := \arg\min_{\mathbf{C}} \sum_{i=1}^{N} \sum_{j=K_i+1}^{L} w_i C^2_{i,j} \quad (20)$$

The mapping is determined by solving the following equation:

$$-\frac{1}{2}\mathbf{y}^T \mathbf{W}_A \mathbf{y} + \mathbf{b}_A{}^T \mathbf{y} = \sum_{i=1}^{N} \sum_{j=K_i+1}^{L} w_i C^2_{i,j} \quad (21)$$

The solution is the following:

$$\mathbf{b}_A = \mathbf{0}_{JL \times 1}, \quad (22)$$

$$\mathbf{W}_A = -2 \begin{pmatrix} \mathbf{D}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_2 & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{D}_J \end{pmatrix}, \quad (23)$$

where

$$\mathbf{D}_j = \begin{pmatrix} \mathbf{0}_{K_j \times K_j} & \mathbf{0}_{K_j \times L-K_j} \\ \mathbf{0}_{L-K_j \times K_j} & w_j * \mathbf{I}_{L-K_j \times L-K_j} \end{pmatrix} \in \mathbb{R}^{L \times L}. \quad (24)$$

The first constraint can be expressed as

$$-\frac{1}{2}\mathbf{y}^T\mathbf{W}_B\mathbf{y}+\mathbf{b}_B{}^T\mathbf{y}=\sum_{i=1}^{N}\left(\left(\sum_{j=1}^{L}C_{i,j}\right)-x_i\right)^2, \tag{25}$$

where the solution is given by

$$\mathbf{b}_B=2\begin{pmatrix}x_{1_{1\times L}} & x_{2_{1\times L}} & \cdots & x_{J_{1\times L}}\end{pmatrix} \tag{26}$$

$$\mathbf{W}_B=-2\begin{pmatrix}\mathbf{1}_{L\times L} & \mathbf{0} & \cdots & \mathbf{0}\\ \mathbf{0} & \mathbf{1}_{L\times L} & \cdots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1}_{L\times L}\end{pmatrix}. \tag{27}$$

The second constraint can be mapped as

$$-\frac{1}{2}\mathbf{y}^T\mathbf{W}_C\mathbf{y}+\mathbf{b}_C{}^T\mathbf{y}=\sum_{i=1}^{M}\left(\left(\sum_{j=1}^{N}C_{i,j}\right)-V\right)^2, \tag{28}$$

whose solution is

$$\mathbf{b}_C=\left[\mathbf{V}_{M\times1},\mathbf{0}_{L-M\times1},\mathbf{V}_{M\times1},\mathbf{0}_{L-M\times1},\ldots,\mathbf{V}_{M\times1},\mathbf{0}_{L-M\times1}\right], \tag{29}$$

$$\mathbf{W}_C=-2\begin{pmatrix}\mathbf{D} & \mathbf{D} & \cdots & \mathbf{D}\\ \mathbf{D} & \mathbf{D} & \cdots & \mathbf{D}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{D} & \mathbf{D} & \cdots & \mathbf{D}\end{pmatrix}, \tag{30}$$

where

$$\mathbf{D}=\begin{pmatrix}\mathbf{I}_{M\times M} & \mathbf{0}_{M\times L-M}\\ \mathbf{0}_{L-M\times M} & \mathbf{0}_{L-M\times L-M}\end{pmatrix}. \tag{31}$$

II.2. I have applied the HNN method for optimization and found an approximate solution to the TWT scheduling problem in polynomial time. I have demonstrated that the HNN solution outperforms simpler heuristics such as EDD, WSPT, LBS, random and LWPF on a large set of randomly generated TWT problems.

A frequently used powerful heuristic algorithm to solve quadratic optimization problems is the Hopfield Neural Network (HNN). This neural network is described by the following state transition rule:

$$\mathbf{y}_i(k+1) = \operatorname{sgn}\left(\sum_{j=1}^{N} \hat{W}_{ij} y_j(k) - \hat{b}_i\right), i = \operatorname{mod}_N k, \tag{32}$$

where

$$\begin{aligned} \mathbf{d} &= -\operatorname{diag}(\mathbf{W}) \\ \mathbf{W} &= -\mathbf{W} - \operatorname{diag}(\mathbf{d}) \\ \mathbf{b} &= \mathbf{b} - \frac{1}{2}\mathbf{d}. \end{aligned} \tag{33}$$

Using the Lyapunov method, Hopfield [24] proved that HNN converges to its fix point, as a consequence HNN minimizes a quadratic Lyapunov function:

$$\mathcal{L}(\mathbf{y}) := -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \hat{W}_{ij} y_i y_j + \sum_{i=1}^{N} y_i \hat{b}_i = -\frac{1}{2}\mathbf{y}^T \hat{\mathbf{W}} \mathbf{y} + \hat{\mathbf{b}}^T \mathbf{y}. \tag{34}$$

Thus, HNN is able to solve combinatorial optimization problems in polynomial time under special conditions as it has been demonstrated in [29] and [30].

I have done a full implementation of this in the MATLAB simulation environment and I have tested the effectiveness of this method against the benchmark methods found in the literature (EDD, SWPT, LBS, LWPF) over a large set of randomly generated problems of varying sizes. The results are depicted in the following figure:

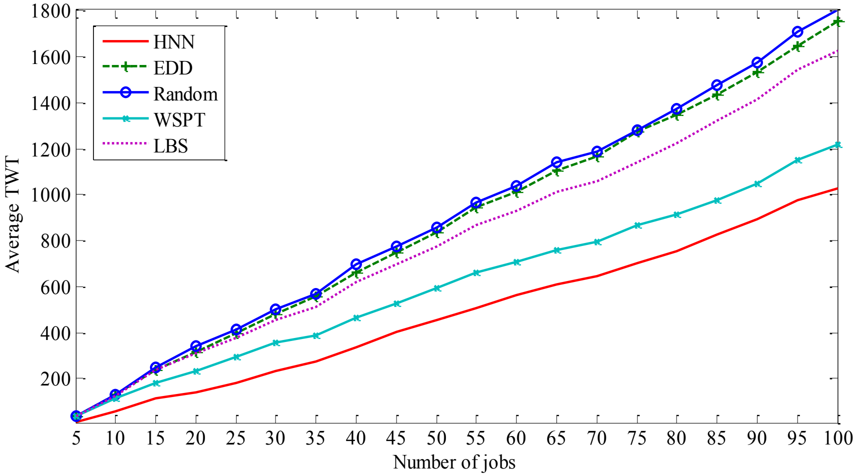

Figure 6: Average TWT produced by each heuristic over randomly generated problems depicted as a function of the number of jobs in the problem.

II.3. I have further improved the HNN performance for the TWT problem by intelligent selections of the starting point (SHNN) and considering perturbations to these starting points (PSHNN).

A block diagram depicting the Perturbed smart Hopfield neural network (PSHNN) method is shown below:

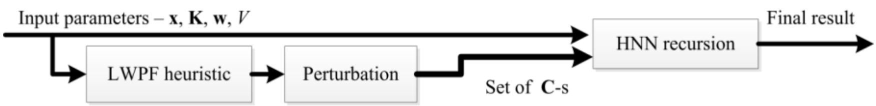

Figure 7: Block diagram of the PSHNN method.

I have found that the PSHNN method consistently outperforms the other methods over a large number of randomly generated TWT problems, as shown in the following figure.

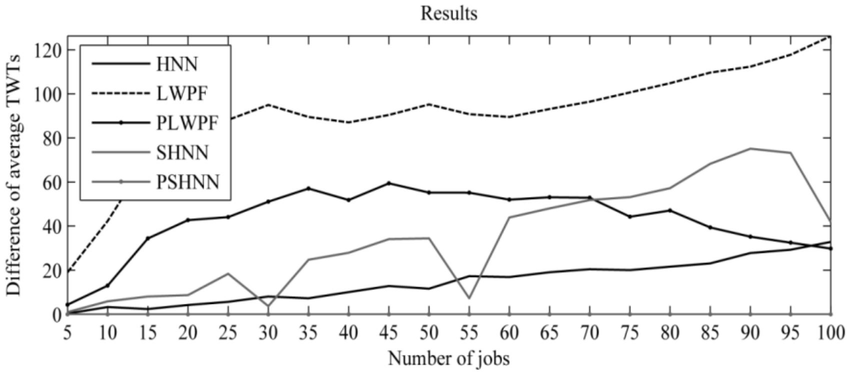

Figure 8: The difference between the average TWT produced by each heuristic and the average TWT produced by the best method, PSHNN over 100 randomly generated problems for each problem size.

As proof of concept, I applied these methods to a specific, real world, large scale scheduling problem involving over 500 jobs for computing risk sensitivities of interest rate derivative portfolios overnight at Morgan Stanley, one of the largest financial institutions in the world. The below table summarizes the results: PSHNN outperforms the next best method, PLWPF by 5%.

Weight	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**SUM**	**Increment to PSHNN**
PSHNN	4401	11116	4020	1620	1092	8	0	0	22 257	0%
PLWPF	3513	9624	5130	1788	490	312	2304	190	23 351	5%
HNN	4404	11040	4735	1824	1092	456	468	0	24 019	7%
LWPF	4404	11140	5470	2472	1183	40	0	0	24 709	10%
EDD	4401	9940	1770	636	1134	464	22752	1430	42 527	48%

Figure 9: Table of total weighted tardiness for jobs of each weight, provided by the different methods for the scheduling problem arising at Morgan Stanley.

4. Conclusions and applications of the results

In this dissertation I have examined two important areas of computational finance: optimal portfolio selection and optimal resource scheduling. I have selected an NP hard open problem within each area and explored polynomial time heuristic methods to provide fast solutions which outperform other benchmark methods. In each case, I have managed to come up with novel approaches which

- produce results which are superior to other heuristics found in the literature,
- have low computational complexity, can be evaluated in polynomial order of time
- have practical runtime characteristics which make them applicable in real world settings

Furthermore, I managed to make a number of other contributions to the solution of each problem. In the case of the sparse, mean reverting portfolio selection I have made significant improvements to the parameter estimation of the VAR(1) and Ornstein-Uhlenbeck processes. In the TWT scheduling problem I have proven that it can be converted to a quadratic optimization problem via a nontrivial matrix algebraic mapping. I have also improved upon the standard HNN method by introducing random perturbations to an intelligently selected initial point (PSHNN method).

Considering the above results, I have achieved the aims of the dissertation.

Finally, in each case I have implemented a proof of concept and have run extensive simulations on both synthetic and real world data. The results on real world data are convincing in each case, having demonstrated trading gains of 34% on historical S&P 500 US stock data using convergence trading of mean reverting portfolios and a 10% TWT improvement over the next best heuristic found in the literature, in the case of a real world large scheduling problem arising at Morgan Stanley.

5. Summary

The dissertation has provided new contributions to computational finance by introducing novel algorithms for a class of combinatorial optimization problems including sparse portfolio optimization (maximizing predictability subject to cardinality constraints) and schedule optimization (minimizing total weighted tardiness subject to capacity constraints).

The achievements of numerical tests on real-world problems are presented in the next table:

Field	*Real world problem*	*Average. performance of traditional approaches*	*Average performance of the proposed new method*	*Impact on computational finance (improvement in percentage)*
Portfolio optimization	Convergence trading on US S&P 500 stock data	11.6% (S&P 500 index return)	34%	22.4%
Schedule optimization	Morgan Stanley overnight scheduling problem	24709 (LWPF performance)	22257 (PSHNN performance)	10%

6. Acknowledgements

I would like to thank my supervisor for his ongoing guidance and direction for the research work. I would also like to thank Kalman Tornai, Robert Sipos, Farhad Kia, dr. Gabor Jeney and dr. Antal Ivanyi for useful discussions and collaboration on some of the problems presented in the theses. I would like to thank Morgan Stanley, my employer, for allowing me to take a leave of absence to focus on this research and also provide data for proving the viability of the methods in real-world applications. Finally, I would like to thank my wife, Ildikó and my entire family for supporting me in completing this work.

7. Publications of the author

Journal Publications [19 points]

1. Fogarasi, N., Levendovszky, J. (2012) A simplified approach to parameter estimation and selection of sparse, mean reverting portfolios. *Periodica Polytechnica,* 56/1, 21-28. [4 points]

2. Fogarasi, N., Levendovszky, J. (2012) Improved parameter estimation and simple trading algorithm for sparse, mean-reverting portfolios. *Annales Univ. Sci. Budapest., Sect. Comp.*, 37, 121-144. [4 points]

3. Fogarasi, N., Tornai, K., & Levendovszky, J. (2012) A novel Hopfield neural network approach for minimizing total weighted tardiness of jobs scheduled on identical machines. *Acta Univ. Sapientiae, Informatica*, 4/1, 48-66. [6/2=3 points]

4. Tornai, K., Fogarasi, N., & Levendovszky, J. (2013) Improvements to the Hopfield neural network solution to the total weighted tardiness scheduling problem. *Periodica Polytechnica*, 57/2, 57-64. [4/2=2 points]

5. Fogarasi, N., Levendovszky, J. (2013) Sparse, mean reverting portfolio selection using simulated annealing. *Algorithmic Finance,* 2/3-4, 197-211. [6 points]

Conference Presentations

6. Fogarasi, N., Levendovszky, J. (2012) Combinatorial methods for solving the generalized eigenvalue problem with cardinality constraint for mean reverting trading. *9th Joint Conf. on Math and Comp. Sci. February 2012 Siófok, Hungary*

Technical Reports

7. Fogarasi, N. (2003) Option pricing primer. *Technical Report*
8. Fogarasi, N. (2003) Non-gaussian option pricing. *Technical Report*
9. Fogarasi, N. (2003) Option pricing using neural networks. *Technical Report*

8. Publications connected to the dissertation

10. Akyol, D.E., & Bayhan, G.M. (2008) Multi-machine earliness and tardiness scheduling problem: an interconnected neural network approach. *International Journal of Advanced Manufacturing Technology*, Springer-Verlag London Limited 37:576-588
11. Armentano, V.A., & Yamashita, D.S. (2000) Taboo search for scheduling on identical parallel machines to minimize mean tardiness. *Journal of Intelligent Manufacturing*, 11, 453-460.
12. Azizoglu, M., & Kirca, O. (1998) Tardiness minimization on parallel machines. *International Journal of Production Economics*, 55, 163-168.
13. Banerjee, O., El Ghaoui, L., & d'Aspermont A. (2008) Model Selection Through Sparse Maximum Likelihood Estimation. *Journal of Machine Learning Research*, 9: 485-516
14. Biskup, D., Herrman, J., & Gupta, J.N.D. (2008) Scheduling identical parallel machines to minimize total tardiness. *International Journal of Production Economics*, 115, 134-142
15. Box, G.E., & Tiao, G.C. (1977) A canonical analysis of multiple time series. *Biometrika* 64(2), 355-365
16. Brucker, P. (2007) *Scheduling Algorithms*. Fifth Edition, New York: Springer
17. D'Aspermont, A, Banerjee, O., & El Ghaoui, L.(2008) First-Order Methods for Sparce Covariance Selection. *SIAM Journal on Matrix Analysis and its Applications*, 30(1) 56-66
18. D'Aspremont, A.(2011) Identifying small mean-reverting portfolios. *Quantitative Finance*, 11:3, 351-364
19. Du, J. & Leung, J. Y.-T. (1990) Minimizing total tardiness on one machine is NP-hard. *Math. Oper. Res.*, 15(3), 483-495.
20. Dogramaci, A., & Surkis, J. (1979) Evaluation of a heuristic for scheduling independent jobs on parallel identical processors. *Management Science*, 25(12), 1208-1216.
21. Fama, E., & French, K.(1988) Permanent and Temporary Components of Stock Prices. *The Journal of Political Economy* 96(2), 246-273
22. Grant, Jeremy (2010) High-frequency trading: Up against a bandsaw. *Financial Times* 2010 Sep 2.
23. Guinet, A. (1995) Scheduling independent jobs on uniform parallel machines to minimize tardiness criteria. *Journal of Intelligent Manufacturing*, 6, 95-103.
24. Hopfield, J. J. (1982) Neural networks and physical systems with emergent collective computational abilities in *Proceedings of the National Academy of Sciences of the USA*

25. Koulamas, C.P. (1994) The total tardiness problem: review and extensions. *Operations Research,* 42, 1025-1041.
26. Lawler, E. L. (1977) A pseudopolynomial algorithm for sequencing jobs to minimize total tardiness. *Annals of Discrete Mathematics*., 1, 331-342
27. Lawler, E. L. (1981) Preemptive scheduling of precedence-constrained jobs on parallel machines, deterministic and stochastic scheduling. *Proceedings of the NATO Advanced Study and Research Institute on Theoretical Approaches to Scheduling Problems*
28. Maheswaran, R., Ponnambalam, S.G., Nithin S. D., & Ramkumar, A.S. (2004) Hopfield Neural Network Approach for Single Machine Scheduling Problem. *Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems*
29. Mandziuk, J. (1996) Solving the Travelling Salesman Problem with a Hopfield - type neural network. *Demonstratio Mathematica*, Vol. 29(1)
30. Mandziuk, J.,& Macukow, B. (1992) A neural network designed to solve the N-Queens Problem. *Biological Cybernetics,* Vol 66(4)
31. Manzan, S. (2007) Nonlinear Mean Reversion in Stock Prices. *Quantitative and Qualitative Analysis in Social Sciences*, 1(3), 1-20
32. Markowitz, H. (1952) Portfolio Selection. *The Journal of Finance* 7:1,77-91
33. Martel, C. (1982) Preemptive Scheduling with Releast Times. *Deadlines and Due Dates Journal of the Association for Computing Machinery*. Vol 29, No 3, 812-829
34. Natarjan, B.K. (1995) Sparse approximate solutions to linear systems. *SIAM J. Comput.* 24(2), 227-234
35. Pinedo, M. (2008) *Scheduling – Theory, Algorithms and Systems*. Third Edition, New York: Springer
36. Poterba, J.M., & Summers, L.H. (1988) Mean reversion in stock prices: Evidence and implications. *Journal of Financial Economics* 22(1), 27-59
37. Rachamadugu, R.M.V., & Morton, T.E. (1983) Myopic heuristics for the weighted tardiness problem on identical parallel machines.*Tech Report CMU-RI-TR-83-17,* Carnegie-Melon University, the Robotics Institute.
38. Rothman, A., Bickel, P., Levina, E., & Zhu, J. (2008). Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics,* 2: 494-515
39. Sahni, S. (1974) Computationally related problems. *SIAM Journal on Computing*, 3.
40. Sahni S. (1979) Preemprive Scheduling with Due Dates. *Operations Research*, 27(5), 925-934
41. Sen, T., Sulek, J.M., & Dileepan, P. (2003) Static scheduling research to minimize weighted and unweighted tardiness: A state-of-the-art survey. *International Journal of Production Economics*, 83, 1-12.